



Contribution des pyramides irrégulières en segmentation d'images multirésolution

Pascal Bertolino

► To cite this version:

Pascal Bertolino. Contribution des pyramides irrégulières en segmentation d'images multirésolution. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 1995. Français. NNT: . tel-00005028v2

HAL Id: tel-00005028

<https://theses.hal.science/tel-00005028v2>

Submitted on 12 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée par

Pascal Bertolino

pour obtenir le titre de
DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE
(Arrêté ministériel du 30.3.1992)

Spécialité : Informatique

CONTRIBUTION DES PYRAMIDES IRRÉGULIÈRES EN SEGMENTATION D'IMAGES MULTIRÉSOLUTION

Date de soutenance : 30 Novembre 1995

Composition du Jury :

Alain Chéhikian	PRESIDENT
Jean Michel Jolion	RAPPORTEUR
Jean Pierre Cocquerez	RAPPORTEUR
Annick Montanvert	EXAMINATEUR
Jean Marc Chassery	EXAMINATEUR
Philippe Bolon	EXAMINATEUR

Thèse préparée au sein du Laboratoire **TIMC - Institut IMAG**

à Sylvie

*à mes enfants Alison
Stevie
Jimmy
Magali*

*à ma mère
à la mémoire de mon père*

Remerciements

Je tiens tout d'abord à remercier profondément ma directrice de recherche et néanmoins amie Madame Annick Montanvert, Professeur à l'ENS de Lyon. Elle m'a donné l'occasion de travailler sur un sujet passionnant, m'a apporté ses idées originales lors de nos nombreuses discussions. Son optimisme sans faille m'a constamment soutenu et m'a permis de bien terminer cette thèse. Je la remercie également pour la confiance qu'elle m'a toujours portée et pour la liberté scientifique dont j'ai disposée.

Merci à Monsieur Jean-Marc Chassery, Directeur de recherche CNRS, directeur de l'équipe INFODIS pour m'avoir accepté au sein de son groupe et pour avoir bien voulu être membre de ce jury. Qu'à travers lui, soit remercié le Groupement de Recherche TDSI du CNRS qui m'a soutenu financièrement pendant une partie de ma thèse.

Je remercie Monsieur Alain Chéhikian, Professeur à l'INP de Grenoble pour l'intérêt qu'il a porté à mon travail, dans son rôle de président du jury de cette thèse et pour ses remarques constructives.

Merci à Monsieur Philippe Bolon, Professeur à l'Université de Savoie pour avoir accepté d'être membre de mon jury.

Je remercie Messieurs Jean-Pierre Cocquerez, Professeur à l'ENSEA de Cergy-Pontoise et Jean-Michel Jolion, Professeur à l'INSA de Lyon, d'avoir eu la lourde tâche d'être rapporteurs. Je suis honoré d'avoir été jugé par des spécialistes du domaine que j'ai traité.

Merci à Monsieur Gérard Brugal, Professeur à l'Université Joseph Fourier de Grenoble, pour m'avoir apporté son soutien lors de mon passage dans son équipe RFMQ, et pour m'avoir permis de participer pendant un an et demi au projet européen IMPACT. Qu'avec lui soit remercié Monsieur Christopher Sowter, Docteur au St Bartholemew's Hospital de Londres avec qui j'ai beaucoup appris, lors de cette thèse.

Merci à Monsieur Walter G. Kropatsch, Professeur à la Technische Universität de Vienne, et directeur du PRIP, pour son enthousiasme dans nos échanges d'idées.

Merci à Monsieur Guy Bourrel, Ingénieur CNRS, grand maître de nos ressources matérielles et logicielles, qui résoud les problèmes en temps réel et avec beaucoup de gentillesse.

Je tiens à remercier tout particulièrement mon ami Gilson Braviano à la fois pour avoir incarné le soleil du Brésil au cœur des Alpes et pour nos discussions et son savoir sur la théorie des graphes.

Merci à mes voisins de bureau : Étienne Bertin, pour sa disponibilité et les idées qu'il m'a apportées ; Franck Davoine, pour sa gentillesse et sa vivacité d'esprit ; Jean-Baptiste Sibarita pour sa sympathie. Merci à Marc Salotti qui partage la même passion que moi pour la segmentation d'images, et avec qui j'ai beaucoup discuté.

Je ne voudrais oublier aucun des collègues (anciens ou actuels) de l'ex RFMQ-CERMO et de l'Institut Albert Bonniot, parmi lesquels les informaticiens (équipes INFODIS et SIC), les biologistes (nul n'est parfait) (équipes DyOGen et RMFQ), et les secrétaires, qui font régner une ambiance agréable, propice aux travaux de recherche.

Enfin, merci à toi, Sylvie, qui m'a toujours épaulé efficacement, au labo comme à la maison. Merci à vous les enfants, d'avoir été aussi adorables et patients.

Résumé

Cette thèse traite des apports de la structure pyramidale irrégulière en segmentation multirésolution d'images en niveaux de gris.

Une présentation des techniques de multirésolution (essentiellement pyramidales) pour le traitement des images et notamment pour la segmentation est faite dans un premier temps.

Dans un second temps, le document décrit des techniques de segmentation d'image par approches 'région' puis par approches 'contour'.

Une troisième partie traite de la modélisation de la pyramide par les graphes, puis détaille précisément sa construction. Cette étude met en évidence certaines faiblesses de l'approche.

Afin d'améliorer les résultats, nous proposons une coopération de processus unissant l'approche région de la structure pyramidale à une approche contour représentée soit par un détecteur de maximums locaux, soit par un détecteur stochastique mis en œuvre à cet effet.

Par la suite, nous développons deux approches qui permettent d'obtenir des pyramides d'images segmentées multirésolution. La première permet, à l'aide des différents niveaux d'une pyramide irrégulière, de fournir rapidement un nombre important de segmentations dérivées, chacune à une résolution différente. La seconde, fondée sur la notion de décimation étendue, génère directement une pyramide d'images à différentes résolutions de détails.

Enfin, quelques éléments concernant l'étude des textures sont proposés. Des résultats expérimentaux sont fournis tout au long du document. Un comparatif des méthodes développées est effectué sur des images type.

Abstract

This thesis shows the contributions of the irregular pyramidal structure within the field of grey level image multiresolution segmentation.

First, multiresolution techniques are reviewed. Then, several image segmentation techniques are explored, involving either region or contour approaches. Thirdly, irregular pyramid construction and its use are thoroughly described, including its flaws.

In order to improve the relevance of this approach, a cooperative process combining the pyramidal structure and either a classical edge operator or a stochastic edge detector (specially developed) is proposed.

Finally, two new methods that provide multiresolution segmented image pyramids are proposed:

The first method provides, by using the different levels of an irregular pyramid, a large number of derived segmentations, each one at a particular resolution.

The second method, based on the wide decimation, directly generates a real image pyramid in which each level contains details at a particular resolution.

Experimental results are given all along the document. A comparison among the proposed methods is performed on test images.

Notations

k	Niveau d'une pyramide
s	Sommet d'un graphe
(s_i, s_j) ..	Arête reliant les sommets s_i et s_j
$G(S, A)$..	Graphe non orienté composé de sommets S et d'arêtes A
$G'(S, A')$	Graphe orienté composé de sommets S et d'arcs A'
$G[k]$	Graphe de niveau k
$S[k]$	Ensemble des sommets d'un graphe de niveau k
$A[k]$	Ensemble des arêtes d'un graphe de niveau k
$A'[k]$	Ensemble des arcs d'un graphe de niveau k
$s[k]$	Sommet d'un graphe de niveau k
$V(s)$	Ensemble des sommets voisins du sommet s
$\sigma(s)$	Ecart type du niveau de gris du sommet s
$g(s)$	Niveau de gris moyen du sommet s
$d(s)$	Degré du sommet s
N	Dimension (largeur et hauteur) d'une image carrée
T	Seuil global
$T(s)$	Seuil local au sommet s
$Card(X)$	Cardinal de l'ensemble X
\leftarrow	Signe de l'affectation

Table des matières

1	Introduction	1
1.1	Introduction générale	2
1.1.1	L'image et son traitement	2
1.1.2	Le problème de la segmentation	2
1.2	Objectifs de la thèse	3
1.3	Structure du document	4
2	Multirésolution pour l'image	7
2.1	Modèles multirésolution	8
2.1.1	Multirésolution par réduction	8
2.1.2	Multirésolution par simplification	9
2.1.3	Calcul de paramètres	11
2.1.4	Complexité algorithmique	12
2.1.5	Machines parallèles	12
2.2	Pyramides passe-bas et Laplaciennes	13
2.2.1	Convolution et sous-échantillonnage	14
2.2.2	Pyramide Gaussienne	14
2.2.3	Pyramide Laplacienne	16
2.2.4	<i>Scale-space</i>	18
2.3	Pyramides géométriques	19
2.3.1	Constructions <i>Bottom-up</i> et <i>Top-down</i>	19
2.3.2	<i>Quadtree</i>	20
2.3.3	Polygonisation de Voronoï	23
2.3.4	Triangulation de Delaunay	25
2.3.5	Autres modèles géométriques	26
2.3.6	Pyramides liées	27
2.4	Pyramides irrégulières	29
2.4.1	Notions de pyramides de graphes	29
2.4.2	Pyramides de régions	29
2.4.3	Pyramides de contours	31
2.4.4	Pyramides duales	31
2.5	Critique des pyramides	32
2.5.1	Avantages	33

2.5.2	Inconvénients	33
2.6	Conclusion	34
3	Techniques de Segmentation	35
3.1	Problématique	36
3.1.1	Segmentations bas et haut niveau	36
3.1.2	But recherché	36
3.2	Approches ‘régions’	37
3.2.1	Formalisme	38
3.2.2	Classification	39
3.2.3	Croissance de région et agrégation	39
3.2.4	Morphologie mathématique	40
3.3	Approches ‘contours’	41
3.3.1	Approches dérivatives	42
3.3.2	Autres approches	45
3.4	Approches coopératives	48
3.4.1	Coopération par contrainte	50
3.4.2	Coopération dynamique	51
3.5	Conclusion	52
4	Pyramide irrégulière	53
4.1	Principes et terminologie	54
4.1.1	Principes pyramidaux	54
4.1.2	Graphes	56
4.1.3	Sous-graphes	57
4.2	Construction	59
4.2.1	De l’image au graphe	59
4.2.2	Structure de données	60
4.2.3	Algorithme général de construction	61
4.2.4	Construction du graphe d’adjacence original	63
4.2.5	Contraction du graphe d’adjacence	63
4.2.6	Construction du graphe de similarité	69
4.2.7	Allocation des sommets non-survivants	74
4.2.8	Mise à jour des sommets survivants	75
4.2.9	Reconstruction du graphe d’adjacence	76
4.2.10	Du graphe à l’image	76
4.3	Multipyramides et consensus	76
4.4	Implantation	78
4.4.1	Structure plate	79
4.4.2	Structure dynamique	79
4.5	Expérimentations	80
4.5.1	Convergence	80
4.5.2	Décimation	82
4.5.3	Influence du seuil global	83

4.5.4	Degré des sommets	84
4.5.5	Influence des tirages aléatoires	85
4.5.6	Résultats expérimentaux	86
4.6	Discussion et conclusion	88
5	Coopération régions-contours	91
5.1	Principes de coopération	92
5.2	Coopération réduite	92
5.3	Coopération étendue	93
5.3.1	Pondération du graphe d'adjacence	94
5.3.2	Mise à jour du graphe de similarité	94
5.3.3	Mise à jour des pondérations	95
5.4	Utilisation de détecteurs existants	96
5.4.1	Recherche des sous-maximums locaux	96
5.4.2	Recherche des éléments de contour	96
5.5	Détection de contours par lancer de segments	97
5.5.1	Opérateur 1D <i>vs</i> opérateurs 2D	98
5.5.2	Principe du détecteur	99
5.5.3	Algorithme	101
5.5.4	Critère de discontinuité adaptatif	102
5.5.5	Prise en compte du bruit	103
5.6	Application aux images de profondeur	103
5.7	Discussion et conclusion	106
6	Aspects multirésolution	107
6.1	Problématique	108
6.1.1	Terminologie	108
6.1.2	Le cas des méthodes de segmentation	108
6.1.3	Le cas de la pyramide irrégulière	108
6.1.4	Éléments de multirésolution	109
6.2	Stratégie <i>top down</i>	111
6.2.1	Pyramide et arborescence	111
6.2.2	Récurtivité de la division irrégulière	111
6.2.3	Traitement <i>Top-down</i> multirésolution	112
6.2.4	Paramètres d'échelle	113
6.2.5	Nouvelle structure de données	114
6.2.6	Résultats expérimentaux	116
6.3	Pyramide multirésolution	117
6.3.1	Décimation d'ordre étendu	117
6.3.2	Extraction de composantes	122
6.3.3	De la pyramide stochastique à la pyramide adaptative	124
6.3.4	Enrichissement des composantes	125
6.3.5	Vers une pyramide d'images multirésolution	126
6.3.6	Résultats expérimentaux	127

6.4	Le cas des images texturées	128
6.4.1	Définition et classification	128
6.4.2	Pyramide irrégulière et pavage de Voronoï	129
6.4.3	Pyramide irrégulière et topologie	130
6.5	Discussion et conclusion	130
7	Résultats comparatifs	135
7.1	Critères d'évaluation	136
7.2	Description des images	136
7.3	Résultats et commentaires	137
8	Conclusion	149
8.1	Bilan	150
8.2	Perspectives	151
	Bibliographie	152
	Index	162

Table des figures

1	Pyramide de Djoser	1
2	Une image à résolutions spatiales décroissantes	9
3	Une image à différentes résolutions de détails	10
4	Autre exemple de multirésolution	10
5	Quelle est la meilleure résolution pour une segmentation?	10
6	Pyramide Gaussienne et pyramides dérivées	16
7	Principe de découpage récursif du <i>quadtree</i>	21
8	Différents partitionnement géométriques de l'image muscle	22
9	Structure d'une pyramide quaternaire	22
10	Partitionnements de Voronoï et de Delaunay	24
11	Structure de données de Voronoï	24
12	Pyramide à base hexagonale	26
13	Pyramide et taux de réduction	26
14	Structure d'une pyramide liée	27
15	Principe d'équicontribution de la pyramide de Peleg	29
16	Pyramide irrégulière	30
17	Graphe normal <i>vs</i> graphe dual	32
18	Importance de la connaissance <i>a priori</i> en segmentation	37
19	Approches 'région' et 'contour' pour segmenter une image	38
20	Segmentation par classification des pixels	39
21	Principe d'obtention de la ligne de partage des eaux (<i>lpe</i>)	42
22	Recherche de contours par approche dérivative	44
23	Opérations élémentaires de morphologie mathématique	46
24	Champs de Markov et contours	47
25	Segmentation par contours actifs	49
26	Segmentation par coopération dynamique	52
27	Pyramide irrégulière et liens	55
28	Correspondance entre champs récepteurs et graphes	55
29	Graphes : exemples	57
30	Exemple de contraction d'un graphe d'adjacence	58
31	Illustration de sous-ensembles de sommets	59
32	4 et 8 voisinage	60

33	Réduction de graphe par sous-échantillonnage	65
34	Pyramides de stables maximaux	65
35	Répartition des sommets de stables maximaux	67
36	Pyramide de champs récepteurs	69
37	Graphe de similarité non orienté et stable maximal	70
38	Problème posé par le graphe de similarité orienté	71
39	Graphe de similarité orienté	72
40	Dégradé segmenté correctement avec l'utilisation du seuil local	72
41	Calcul du seuil local de similarité	74
42	Problème posé par les graphes orientés	74
43	Allocation des non-survivants	75
44	Création des adjacences du niveau supérieur	77
45	Carte de contours pondérés pour l'image de consensus	78
46	Convergence des images de consensus	78
47	Principes de la structure pyramidale plate	80
48	Nombre de régions en fonction des niveaux	81
49	Taux de décimation sur un graphe d'adjacence	83
50	Nombre de régions par niveau en fonction du seuil global	84
51	Segmentations pour différents seuils	84
52	Evolution du degré des sommets	85
53	Influence du tirage aléatoire	86
54	Robustesse de la décimation	87
56	Problématique de la pyramide irrégulière sur une image de synthèse	88
55	Pyramide irrégulière de l'image 'muscle'	90
57	Initialisation des pondérations du graphe d'adjacence	94
58	Exemple de mise à jour inter-niveaux des pondérations	95
59	Coopération région-contour avec l'opérateur de Deriche	97
60	Problème d'exhaustivité des éléments de contour	98
61	Relation entre régions et contours	99
62	Exemples de 3 segments	100
63	Segments et profils divers	100
64	Evolution du nombre d'éléments de contours détectés	102
65	Prise en compte du bruit	103
66	Détection stochastique avec prise en compte du bruit	104
67	Coopération région-contour avec l'opérateur stochastique	105
68	Caractéristiques de l'image de profondeur utilisée	105
69	Segmentation sans coopération	106
70	Segmentation avec coopération	106
71	Problématique de la multirésolution de la pyramide irrégulière	109
72	Division récursive des champs récepteurs	112
73	Deux pixels et leurs ascendants successifs	114
74	Multirésolution obtenue par variation du paramètre d'échelle	116

75	Ordre de voisinage. Exemple	117
76	Effet de bord du critère d'unification	119
77	Pyramide à base locale	120
78	Propagation des sommets unificateurs	121
79	Croissance de régions étendue	122
80	Extraction d'entités	123
81	Problème posé par la décimation stochastique. Exemple	124
82	Solution apportée par la décimation adaptative. Exemple	125
83	Segmentation ne comportant que des racines	126
84	Pyramide classique <i>vs</i> pyramide multirésolution	132
85	Pyramide multirésolution obtenue par décimation étendue	133
86	Extraction des tesselles de plusieurs textures	133
87	Histogramme des niveaux de gris des images utilisées pour les tests	138
88	Profil significatifs des images utilisées pour les tests	141
89	Segmentation de l'image 'Aquitaine'	142
90	Segmentation de l'image 'Angiographie'	143
91	Segmentation de l'image 'Bagues'	144
92	Segmentation de l'image 'Couloir'	145
93	Segmentation de l'image 'IRM'	146
94	Segmentation de l'image 'Muscle'	147
95	Pyramides de Guizèh	149

Chapitre 1

Introduction

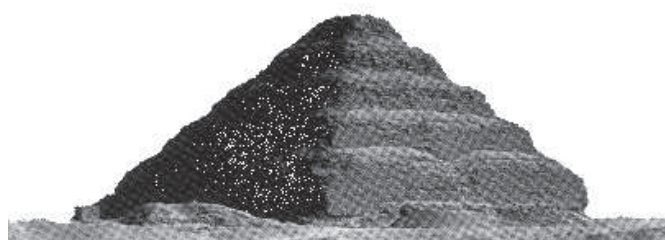


Fig. 1 - Pyramide de Djoser

1.1 Introduction générale

Grâce au récent bond en avant technologique dans le domaine des images numériques et de leur diffusion¹, l'image est devenue un support d'information omniprésent dans notre réalité quotidienne, dans l'industrie et la science.

La capacité de représentation des images permet de développer des systèmes de plus en plus autonomes au prix d'une automatisation et d'une complexité accrues de leur traitement.

Dans ce premier chapitre, nous parlons brièvement des différentes composantes du traitement d'image, et nous posons les problèmes liés à la segmentation.

1.1.1 L'image et son traitement

La segmentation intervient au cœur de la discipline appelée 'analyse ou traitement d'image' [Ros82]. C'est sans doute la clef de voûte de ce domaine, mais également la partie qui pose le plus de problèmes et qui s'avère le plus difficile à automatiser.

Théoriquement, l'analyse d'image est un ensemble d'approches, de méthodes, de techniques et d'outils offerts par les mathématiques et le traitement du signal. Cette panoplie a la volonté de résoudre la majorité des problèmes qui peuvent se présenter lorsqu'il est nécessaire d'extraire et de comprendre de façon automatique les informations présentes dans une image. Ces informations peuvent être significatives pour le système visuel ou pour un domaine d'application particulier.

En pratique, l'analyse d'image est une suite de phases qui doivent être exécutées, depuis la formation de l'image jusqu'à la prise de décision fondée sur son contenu. Certaines de ces phases successives sont souvent étroitement liées et souvent indissociables. Les unes sont obligatoires alors que l'exécution de certaines autres s'avère parfois facultative.

L'analyse d'image automatisée naît avec le premier analyseur d'images en 1963. Elle voit son essor grâce au progrès de l'informatique et à l'apparition des images en niveaux de gris héritées de la recherche spatiale. A l'heure actuelle, l'analyse d'image est une discipline dont les fondements théoriques reposent sur les mathématiques et les fondements pratiques sur l'informatique et l'électronique.

1.1.2 Le problème de la segmentation

La segmentation d'image a pour but d'extraire de l'image un certain nombre d'entités différentes appelées objets ou régions. Ces entités doivent être représentatives du contenu de la scène pour permettre des traitements avals optimaux d'extraction

¹Nous parlons ici d'images naturelles. Le domaine de l'image de synthèse n'est pas abordé dans cet ouvrage.

de caractéristiques, de reconnaissance de forme et d'interprétation. Comment notre système visuel accomplit-il cette tâche? Dans le plus grand secret. Nous ne pouvons qu'estimer qualitativement ce qui paraît important dans son mode de fonctionnement :

- vision globale ;
- focalisation locale ;
- traitements qualitatifs très rapides ;
- traitements quantitatifs lents ;
- perceptions gestaltistes [Spo87][Roc91] ;
- utilité des connaissances sémantiques ;
- multirésolution.

Munie de ces éléments, la segmentation est une tâche facile voire triviale pour notre système visuel. En revanche, elle pose de nombreux problèmes non résolus en analyse automatique. L'explication la plus plausible de cet échec est sans doute une modélisation trop simpliste de l'information contenue dans l'image [Pav92] ainsi qu'une utilisation trop limitée de ces informations.

Quant à nous, plutôt que de considérer l'image comme plusieurs problèmes réunis admettant une unique solution, nous préférons voir un problème qui admet plusieurs solutions. Pour cette raison, nous orientons nos travaux sur la représentation multi-résolution d'une image, représentation qui permettrait, grâce à la segmentation, de reconstituer le contenu d'une image avec plus ou moins de détails.

1.2 Objectifs de la thèse

Cette thèse reprend des travaux récents sur une approche nouvelle et séduisante : la segmentation d'image par pyramide irrégulière. Les résultats obtenus avec la méthode originale montrent son fort potentiel sur de nombreuses images ainsi que certaines faiblesses.

La qualité des résultats obtenus dans certains cas laissent entrevoir la possibilité de déterminer la nature des difficultés rencontrées et les moyens à mettre en œuvre pour les surmonter.

Dans ce document, notre objectif est triple :

- Tout d'abord, il est d'attirer l'attention sur la segmentation d'image et ses différents aspects, à travers un état de l'art que nous avons voulu varié sans avoir aucune prétention d'exhaustivité.

- Ensuite, nous avons voulu pousser plus loin l'approche pyramidale irrégulière, convaincus qu'elle peut donner des résultats au moins aussi bons que les meilleures méthodes existantes. La pyramide irrégulière, de par sa modélisation et son fondement sur les graphes se prête à de nombreux enrichissements. Nous espérons élargir la voie des recherches dans cette direction.
- Enfin, désirons travailler sur des problématiques porteuses telles que la coopération de processus et la multirésolution. Nous pensons que ces techniques tentent de recréer à leur manière le mode de fonctionnement de notre système visuel. Nous sommes conscients que le degré de complexité et la faculté de traitement qualitatif de ce dernier ne peuvent être ni modélisés ni égalés. En revanche, nous sommes convaincus que certains traitements de bas niveau peuvent être améliorés.

Nous espérons que tout au long de cette thèse, le lecteur ressentira cette volonté.

1.3 Structure du document

Ce document peut être découpé en deux parties distinctes :

La première est un tour d'horizon sur les approches, méthodes et philosophies en rapport avec nos centres d'intérêts. Cet état de l'art non exhaustif comporte deux chapitres qui reprennent chacun une composante de notre thématique :

1. Le deuxième chapitre est consacré à une **introduction à la multirésolution** ainsi qu'à un certain nombre de modèles pyramidaux pour le traitement d'image.
2. Le troisième chapitre concerne les deux grandes familles de **segmentation**, dont le but est de trouver respectivement des régions et des contours dans une image en niveaux de gris. Nous étudions de quelle manière ces approches peuvent coopérer afin d'améliorer leurs performances.

La seconde partie concerne l'étude de la méthode utilisée et nos propres travaux de recherche dans le domaine. Elle est découpée en quatre chapitres :

1. Le quatrième chapitre est consacré à la **pyramide irrégulière**, matière de base de notre recherche. Nous réalisons une étude approfondie de sa structure et de son fonctionnement, étayée de résultats expérimentaux. Sur la base d'une critique constructive de ces résultats, nous posons notre problématique et nous apportons des éléments de réponse qui sont développés par la suite.
2. Le cinquième chapitre montre une manière de mettre en place une **coopération régions - contours** pour améliorer les performances de la segmentation. Cette coopération est faite entre le processus de fusion en parallèle de régions

de la pyramide irrégulière d'une part, et un détecteur de contours (classique puis spécifique) d'autre part.

3. Le sixième chapitre présente deux méthodes nouvelles permettant de réaliser de la **segmentation multirésolution**.
4. Le septième chapitre décrit les **résultats** obtenus avec chacune des méthodes sur un jeu d'images tests.

Le dernier chapitre conclut et donne un certain nombre de **perspectives** sur l'utilisation de la pyramide irrégulière en segmentation d'image.

Chapitre 2

Multirésolution pour l'image

Selon l'approche multirésolution, il existe pour chaque entité d'une image, une résolution à laquelle la recherche, l'extraction et la description de cette entité sont les meilleures. Depuis les années 70, de nombreux domaines du traitement d'image bénéficient de travaux utilisant la multirésolution.

Dans ce chapitre, sont exposés tout d'abord, les principes de la multirésolution. Ensuite, nous étudions les relations étroites entre la multirésolution, le parallélisme et les structures pyramidales en traitement d'image de bas niveau. Pour cela, nous rappelons le fonctionnement des différentes structures, leurs domaines d'application et leurs limites.

On remarquera que, dans le domaine de la segmentation, la multirésolution est utilisée en tant que décomposition hiérarchique de l'image et non dans un sens plus orienté vers la description des entités à différentes résolutions sémantiques.

Parmi les modèles multirésolution, la pyramide occupe une place de choix. De nombreuses structures ont été proposées depuis les premiers travaux menés sur le sujet [Kel71] [ST75]. Le lecteur intéressé par la pyramide comme modèle de représentation informatique peut se rapporter à [Jol90]. Des études plus complètes sont faites dans [Kro91a] et [Jol94b].

2.1 Modèles multirésolution

Intuitivement, la multirésolution peut apparaître à notre système visuel de différentes manières :

- réduction de la taille de la scène observée (éloignement) ;
- diminution de la netteté ;
- simplification du contenu.

Sous l'aspect structural, la représentation multirésolution d'une image se présente de la manière suivante : en utilisant un traitement approprié sur une image, on peut obtenir une série d'images 'dérivées' qui contiennent (par l'un des procédés énoncés ci-dessus) de moins en moins d'information. Considérons les éléments de la multirésolution comme un empilement de ces images de résolution décroissante. Cette structure est une pyramide, composée d'une succession de niveaux, où le niveau de la base est l'image originale et les niveaux successifs sont de résolution de plus en plus réduite. Dans la grande majorité des méthodes multirésolution, chaque résolution (ou *niveau*) est obtenue à partir de la résolution précédente et non pas à partir de l'image originale, exception faite du *scale-space* (§ 2.2.4), un cas particulier qui n'est pas considéré comme une technique pyramidale.

La principale motivation des chercheurs pour les structures pyramidales et multirésolution vient qu'elles ont de nombreux points communs avec la structure et le fonctionnement de notre système visuel qui est massivement parallèle :

- Les cônes de reconnaissance de Uhr sont un ensemble de procédures destinées à gérer des pyramides d'images [Uhr87]. Ces cônes consistent en une série de traitements perceptifs (lissage, détection d'éléments de contour primaires, formation de contours plus longs, formation d'objets puis classification).
- Dans [Ros87], l'auteur propose une approche pour reconnaître des objets inattendus dans une scène, en fondant la structure pyramidale sur le comportement humain.
- Le phénomène de la fovéa peut être modélisé avec une structure neuronale rétinienne [Man94].

2.1.1 Multirésolution par réduction

La multirésolution peut être vue sous son aspect spatial : dans ce cas, la taille de l'image diminue et provoque une diminution de la résolution (figure 2). Ce procédé fait intervenir un traitement de sous-échantillonnage, accompagné d'un filtrage qui permet de conserver un maximum d'information malgré la perte provoquée par le sous-échantillonnage (§ 2.2.2).

Les méthodes qui mettent en jeu la multirésolution spatiale ne sont pas des techniques de segmentation à part entière mais permettent généralement de segmenter

des images à une dimension réduite. Ensuite le résultat obtenu est converti à la taille de l'image originale, avec prise en compte des détails perdus lors de la baisse de résolution.

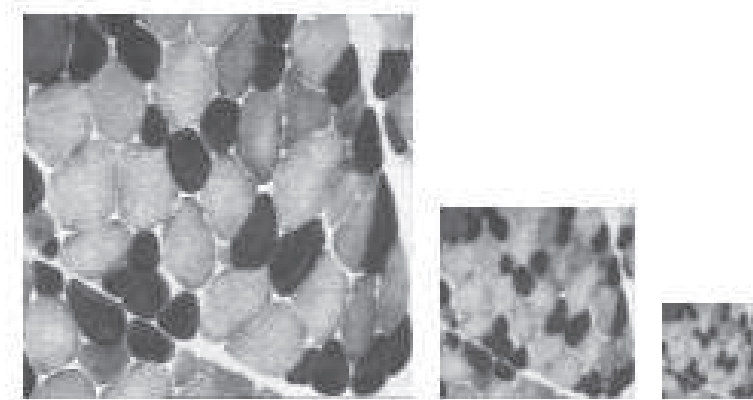


Fig. 2 - Une image à résolutions spatiales décroissantes

Dans ce domaine, les méthodes existantes considèrent l'image comme un signal et proposent des traitements numériques qui approximent au mieux les traitements effectués sur les signaux analogiques.

2.1.2 Multirésolution par simplification

Sous cet aspect, la résolution est liée au détail : les différents niveaux ont tous la taille de l'image initiale. Seul le degré de détail décroît avec la résolution. En pratique, cette représentation est rendue possible par l'utilisation de **champs récepteurs** (§ 4.1.1). En réalité, ces pyramides sont des pyramides doubles, un premier empilement modélisant l'adjacence des régions, et un second empilement représentant la projection des différents niveaux à l'échelle de l'image originale. Avant d'aller plus loin, donnons la définition d'une *cellule*, terme qui sera utilisé par la suite.

Définition 1 Une *cellule* est la composante primitive d'une pyramide. Elle correspond au sommet dans le contexte d'un graphe, au pixel ou à la région dans le contexte d'une image, et au processeur dans le contexte matériel.

Certaines des méthodes qui procèdent par simplification sont des techniques de segmentation multirésolution car chacune des images qu'elles génèrent est une segmentation à part entière (figure 3). Un niveau de la pyramide est alors un ensemble de régions (cellules) dont la forme est souvent liée à la méthode, et la taille est liée au niveau de résolution.

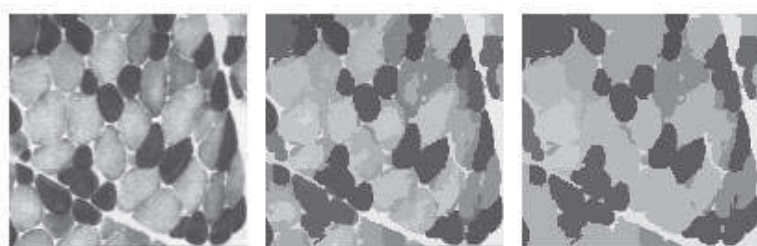


Fig. 3 - Une image à différentes résolutions de détails

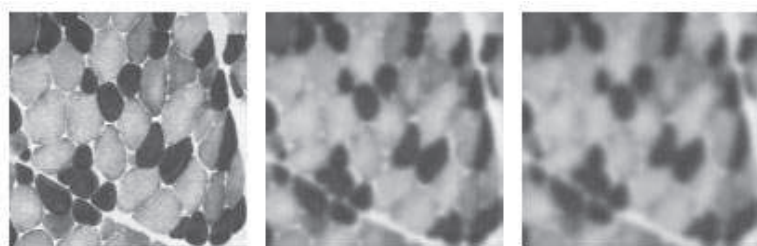


Fig. 4 - Autre exemple de multirésolution

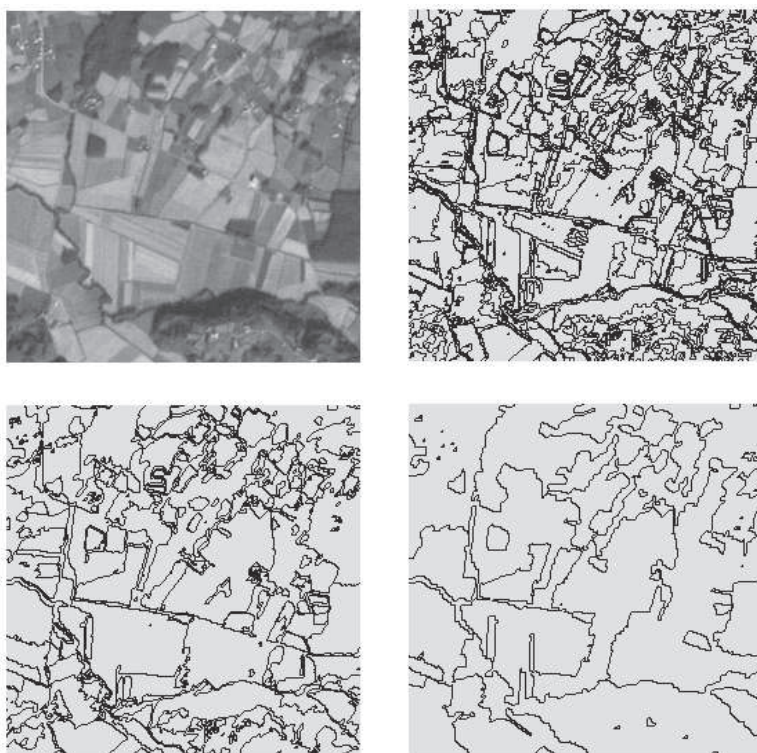


Fig. 5 - Image aérienne. Quelle est la meilleure résolution pour une segmentation ?

D'autres techniques (*scale-space* par exemple (§ 2.2.4)) utilisent le filtrage et résolvent en grande partie les difficultés rencontrées lors de la segmentation (figure 4).

Hormis la question de savoir si une segmentation est bonne ou mauvaise, la multirésolution propose de choisir la meilleure segmentation parmi plusieurs (figure 5), en fonction du degré de détail nécessité par l'application.

Il peut être facile *a posteriori* de déterminer visuellement le niveau 'optimal' de description d'une image. En revanche, ce processus est difficile à automatiser, sauf dans certains cas où le type d'image à traiter est connu : nombre et type d'objets (manufacturés, naturels), éclairage, contraste. Alors, il est possible de prévoir un ordre de grandeur pour le nombre de régions attendu et ainsi de régler au mieux les différents paramètres (seuils) qui conditionnent la perte de résolution.

Le problème est encore plus complexe dès lors que l'image contient des zones ou des objets qui doivent être pris à des résolutions différentes.

2.1.3 Calcul de paramètres

L'extraction de primitives, est un des points forts des approches pyramidales [Ros86]. Celles-ci sont efficaces pour extraire des pixels caractéristiques de l'image ou calculer rapidement des paramètres statistiques relatifs à une image. Cette efficacité est due aux calculs locaux et indépendants qui peuvent y être effectués. A chaque niveau, la valeur d'une cellule (pixel, région ou sommet de graphe) est calculée à partir de ses fils. Ces calculs de paramètres statistiques sont donc effectués de manière récursive. Le plus simple de ces paramètres est la moyenne des pixels d'une image calculée à partir de la somme des niveaux de gris de l'ensemble des pixels. Cela donne, pour un pixel w ayant 4 fils v_i , en utilisant une pyramide quaternaire :

$$g_k(w) = \frac{\sum_{i=1}^4 g_{k-1}(v_i)}{4}$$

La variance des niveaux de gris d'une image peut être calculée simplement en utilisant la formule incrémentale qui nécessite de mémoriser uniquement la somme des niveaux de gris S et la somme de leur carré $S2$:

$$Var_k(w) = \frac{S2_k(w) \Leftrightarrow 4^k S_k(w)}{4^k \Leftrightarrow 1}$$

L'utilisation de cette méthode permet d'obtenir pour ces paramètres une complexité algorithmique en $O(\log_2(N))$ pour une image de taille $N \times N$.

En détection d'objet, la multirésolution permet, sur des machines pyramidales, d'effectuer rapidement les recherches sur des images de faible taille pour ensuite restituer la totalité de l'information relative à l'objet cherché en redescendant dans la pyramide. Cette recherche utilise les pyramides régulières quaternaires avec recouvrements [Bur81b] (§ 2.3.6).

La détection de points brillants [Bla88] est un exemple simple de traitement *bottom up* puis *top down* (§ 2.3.1). Tout d'abord, une pyramide quaternaire de minimums est construite : récursivement, chaque père prend la valeur de son fils ayant la valeur minimale. Ensuite, lorsque l'apex est atteint (sa valeur étant égale au pixel de valeur minimale), une descente dans la pyramide permet, en sélectionnant récursivement le fils de valeur minimale, de trouver le (ou les) pixel(s) recherché(s). Le but final n'est pas, bien entendu, de trouver le pixel le plus brillant (qui peut correspondre à du bruit), mais la région la plus brillante. En utilisant un calcul de moyenne à la place de la recherche de minimum, le pixel extrait a toutes les chances d'appartenir à une région plus claire que les autres.

2.1.4 Complexité algorithmique

La complexité algorithmique mesure, en fonction de la taille de l'image, le nombre de traitements élémentaires appliqués aux composantes de l'image (i.e. les cellules). Prenons l'exemple d'une image carrée $N \times N$ ($N = 2^m$). Une recherche de maximum dans cette image sur une architecture séquentielle est typiquement de l'ordre de $O(N^2)$.

Les architectures parallèles munies d'algorithmes pyramidaux permettent des traitements indépendants effectués en parallèle sur toutes les cellules d'un niveau : théoriquement, N^2 processeurs effectuent chacun le traitement élémentaire une fois. La taille d'une pyramide de type quaternaire (§ 2.3.2) est $\log_2(N) + 1$, la complexité algorithmique associée est donc $O(\log_2(N))$.

Dans le cas des pyramides irrégulières où le rapport du nombre de cellules inter-niveaux n'est pas de 1 pour 4 mais un peu inférieur, la taille de la pyramide reste cependant en $O(\log_2(N))$.

2.1.5 Machines parallèles

Ce paragraphe concerne les analogies qui existent entre les structures de données pyramidales et les machines à architecture parallèle. Cette présentation permet de voir que les approches pyramidales, qui sont souvent défavorisées sur des architectures à mono-processeur, montrent toute leur supériorité lorsqu'elles sont implantées sur machines parallèles. Une machine parallèle pyramidale est formée de trois composantes :

1. Un ensemble de processeurs indépendants possédant leur propre mémoire. Chaque processeur est une cellule de la pyramide.
2. Un contrôleur central dont le rôle est le séquençement des opérations effectuées par les processeurs. Il sert également d'interface avec le monde extérieur. Cette fonction peut être assurée par le processeur situé à l'apex (§ 4.1.1) de la pyramide.

3. Un réseau de communication reliant les différents processeurs entre eux et combinant deux types de liaisons : horizontales et verticales.

Chaque cellule d'un niveau de la pyramide est connectée à ses voisins de même niveau (quatre ou huit). Chaque cellule de niveau k est reliée à des éléments de niveau $k \Leftrightarrow 1$ appelés fils et une ou plusieurs cellules de niveau $k + 1$ appelées pères. La base et l'apex de la pyramide représentent de ce point de vue des cas particuliers.

Plusieurs projets de réalisations matérielles de machines parallèles ont vu le jour, parmi lesquels : PAPIA (Pyramidal Architecture for Parallel Image Analysis) [Can85], SPHINX (Système Pyramidal Hiérarchisé pour le traitement d'Images Numériques) [Mér85]. Chaque élément de ces pyramides est un processeur 1 bit disposant de 128 ou 256 bits de mémoire propre. PAPIA est conçue selon le modèle arborescent quaternaire 4-connexe sans recouvrement (§ 2.3.2). SPHINX est une machine à structure binaire qui traite des images 128×128 .

Des machines non pyramidales dans leur structure simulent des machines pyramidales : la Pyramid Vision Machine [dW85] est constituée de trois modules de mémoire de 512×512 éléments chacun et de deux modules de calcul dont un spécialisé dans les manipulations d'adressage permettant de simuler les différents niveaux de la pyramide. Cette machine traite 30 images 256×240 par seconde. Elle est très bien adaptée pour le suivi d'objets en temps réel. Les mécanismes mis en œuvre reposent sur l'utilisation des pyramides Laplacienne et Gaussienne (§ 2.2).

Une pyramide peut également être simulée sur une architecture de type *array processor*. Tous les niveaux sont stockés dans une seule matrice d'éléments grâce à un jeu d'adresses. La Connection Machine de type SIMD (Single Instruction, Multiple Data-Stream) [Pog88] utilise cette approche. Elle possède 65536 processeurs.

Les machines à base de Transputers telle EGPA (Erlangen General Purpose Array) [Han84] présentent le double avantage des éléments pyramidaux (processeurs 32 bits) et de la mémoire locale importante (4 mégaoctets par processeur). Cependant, cette technologie ne permet pas encore un nombre élevé d'éléments (le réseau T. Node TN364 comporte 64 Transputers T800 et EGPA ne comporte que 4×4 éléments à sa base). Cet inconvénient peut être amoindri si on utilise comme entité de base non pas le pixel mais une sous-image de l'image initiale. Néanmoins, un Transputer ne comporte que quatre liens ce qui complique et limite son utilisation pour construire des pyramides.

La 2D hybrid pyramid vision machine [Li93], comprenant des multiprocesseurs SIMD et des Transputers, permet de faire de la reconnaissance d'objet en temps réel à l'aide d'une structure arborescente binaire effectuant des traitements de bas et de haut niveau.

2.2 Pyramides passe-bas et Laplaciennes

Dans cette partie, nous nous intéressons à une structure de nature particulière qui utilise des concepts de traitement du signal pour filtrer l'image.

Nous introduisons tout d'abord le concept de produit de convolution et de sous-échantillonnage. Nous voyons ensuite comment la multirésolution peut être obtenue, soit en appliquant un masque de convolution de taille fixe sur des images de plus en plus petites, soit en appliquant des masques de convolution de taille croissante sur une image de taille fixe.

2.2.1 Convolution et sous-échantillonnage

Il est toujours possible de réduire successivement la taille d'une image par un facteur quelconque en affectant à chaque élément père une valeur calculée à l'aide de ses fils (plus ou moins nombreux en fonction du taux de contraction (§ 2.3.2) et du recouvrement choisis (§ 2.3.6)) : moyenne, médiane, etc ... Cette méthode, mise au point par [Kel71] en 71, a été la toute première utilisée pour représenter une image entière à différentes résolutions. L'auteur fait de la détection de contours à basse résolution, puis affine leur localisation en revenant sur l'image originale. Dans ce cas, la perte d'information entre deux niveaux successifs est quasi aléatoire et non maîtrisée.

Burt [Bur81a] montre en faisant référence au théorème de Shannon que la perte minimale d'information se fait en accompagnant le sous-échantillonnage par un filtrage passe-bas. Ce filtrage est effectué par un produit de convolution entre les pixels de l'image et un masque de convolution idoine (également appelé *noyau*)¹. Les noyaux utilisés sont des approximations discrétisées de fonctions Gaussiennes, mais leur constitution peut varier en fonction du but recherché.

[Ché92] améliore l'approche en proposant tout d'abord un noyau de filtrage optimal qui donne des images de meilleure qualité et en développant un algorithme de complexité inférieure.

Une étude détaillée de différents noyaux pairs et impairs est faite dans [Kon94].

2.2.2 Pyramide Gaussienne

La pyramide Gaussienne [Bur83] est la structure pyramidale typique puisque la taille des images est physiquement réduite de niveau en niveau (figure 6.a). Elle est utilisée en codage d'image [Huf52] et également en segmentation pour son aspect réel de multirésolution. La pyramide Gaussienne est une séquence d'images dans laquelle chaque image de niveau $k + 1$ a une surface qui représente le quart de celle du niveau k . Pour une image de taille $N \times N$, la pyramide possède donc $\log_2(N) + 1$ niveaux, le niveau 0 étant l'image initiale et le niveau $\log_2(N)$ l'apex constitué d'un seul pixel. Le processus de réduction (procédure REDUCE) est effectué à la fois par un filtrage et par un sous-échantillonnage. Le filtrage de type passe-bas est obtenu par un noyau de convolution dont les valeurs des coefficients discrets approximent

¹Comme cette approche utilise les outils du traitement du signal, la terminologie d'usage ne fait pas référence aux liens 'père-fils'. Néanmoins, il est intéressant de noter l'analogie entre ces liens et le traitement de filtrage-échantillonnage.

Algorithme 1 (*Pyramide Gaussienne*)

```

début
  niveau <- 0;
  image_Gaussienne[niveau] <- image_originale;
  tantque (niveau < n) faire
    début
      image_Gaussienne[niveau+1] <- REDUCE(image_Gaussienne[niveau]);
      image_expanded <- EXPAND[image_Gaussienne[niveau+1]];
      image_Laplacienne[niveau] <- SUBTRACT(image_Gaussienne[niveau],
                                              image_expanded);
      niveau <- niveau+1;
    fin;
  fin.

```

les valeurs d'une courbe Gaussienne.

Le noyau généralement utilisé est de dimension 5×5 . Les coefficients $w(m, n)$ ($-2 \leq m, n \leq 2$) ne peuvent pas être choisis librement et doivent répondre à certaines contraintes relatives à leur séparabilité, leur normalisation, leur symétrie et leur équicontribution [Bur84]. Le traitement qui permet de construire les deux pyramides est systématique et non guidé par le contenu de l'image. L'algorithme 1 montre le traitement d'une image de taille $N \times N$. La pyramide est obtenue par des réductions et des expansions successives des images. Pour chaque pixel de coordonnées (i, j) , la procédure **REDUCE** utilise la formule :

$$G_k(i, j) = \sum_{m=-2}^{m=2} \sum_{n=-2}^{n=2} w(m, n) G_{k-1}(2i + m, 2j + n)$$

La procédure **EXPAND** est la suivante :

$$G_k(i, j) = 4 \sum_{m=-2}^{m=2} \sum_{n=-2}^{n=2} G_{k+1}\left(\frac{i+m}{2}, \frac{j+n}{2}\right)$$

Les pyramides Gaussiennes et Laplaciennes sont à la base de nombreuses approches. Citons quelques exemples :

Les ondelettes [Mal89] ont un fonctionnement assez similaire aux pyramides Gaussienne et Laplacienne : l'image est décomposée en 4 sous-images représentant les détails (hautes fréquences horizontales, verticales et diagonales), ainsi que les basses fréquences. Les 4 pyramides sont obtenues récursivement par un filtrage (convolution et sous-échantillonnage) de l'image basse fréquence obtenue.

Les champs de Markov, grâce aux pyramides, peuvent bénéficier d'une accélération de la convergence : dans un processus *top down*, les champs d'étiquettes obtenus à basse résolution permettent d'initialiser l'algorithme d'estimation au niveau inférieur.

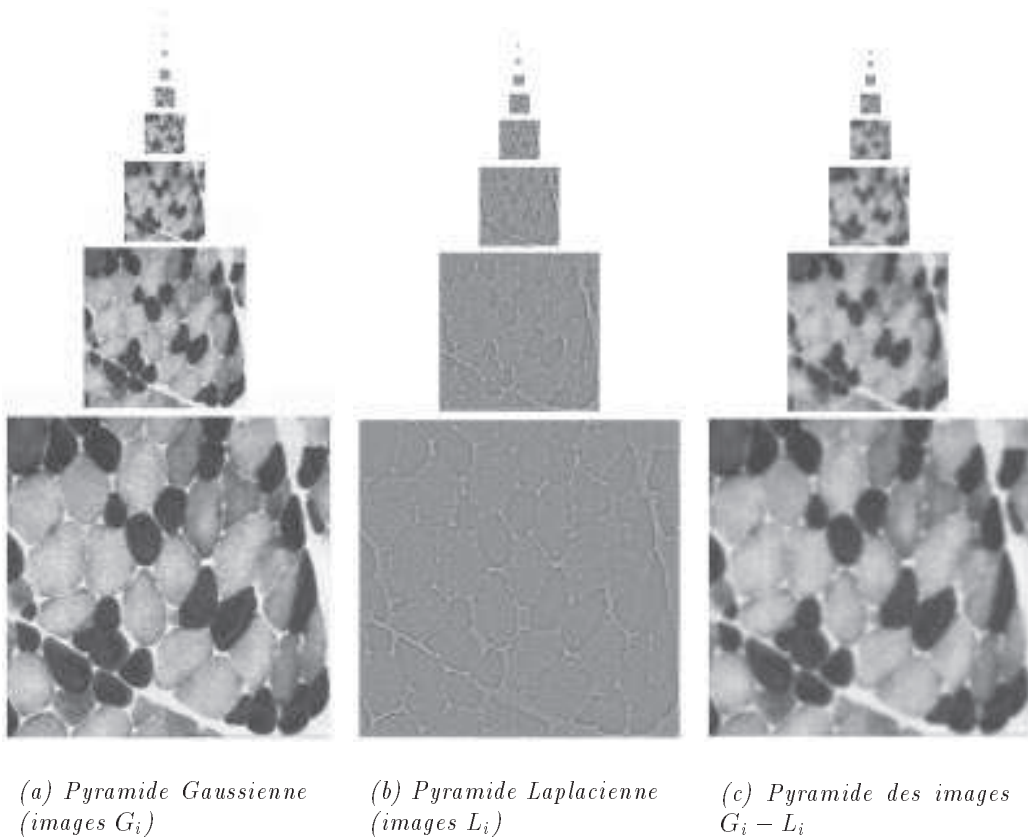


Fig. 6 - *Pyramide Gaussienne et pyramides dérivées*

Comme alternative récente aux diverses améliorations par modification de l'histogramme des niveaux de gris, [Jol94a] propose une méthode qui prend en compte l'aspect topologique du contraste : une pyramide d'images de contraste est tout d'abord construite à partir des pyramides Gaussienne et Laplacienne. Ensuite, chaque niveau de la pyramide améliorée est obtenu en pondérant chaque pixel de l'image Gaussienne correspondante, par un poids plus ou moins grand que 1 selon sa valeur de contraste. L'itération du procédé permet d'obtenir des images binaires représentatives du contenu de l'image.

2.2.3 Pyramide Laplacienne

La pyramide Gaussienne possède son élément dual : la pyramide Laplacienne. Chaque niveau k de cette pyramide est construit en faisant la différence pixel à pixel (procédure SUBTRACT) de deux niveaux successifs k et $k + 1$ de la pyramide Gaussienne (figure 6.b). Cette différence pixel à pixel est rendue possible par une

Algorithme 2 (*Principe de transmission progressive*)

```

début
    calculer tous les  $G_i$  et  $L_i$ ;
    apex =  $\log_2(N)$ ;
    émission / réception  $G_{\text{apex}}$ ;
    Image_traitée =  $G_{\text{apex}}$ ;
    Image_affichée = Image_traitée;
    k <- apex;
    faire k fois
        Image_affichée = EXPAND(image_affichée);

    AFFICHER(image_affichée);
    tantque (qualité(image_affichée) insuffisante) faire
        début
            k <- k-1;
            émission / réception  $L_k$  ;
            Image_traitée <- EXPAND(Image_traitée) +  $L_k$ ;
            Image_affichée = Image_traitée;
            faire k fois
                Image_affichée = EXPAND(image_affichée)
            AFFICHER(image_affichée);
        fin;
fin.

```

expansion du niveau $k + 1$ Gaussien à la taille du niveau k (procédure EXPAND). Le niveau Laplacien k ainsi construit comporte toutes les hautes fréquences (détails) perdues par le passage du niveau k au niveau $k + 1$ Gaussien. La dualité s'exprime :

$$\begin{aligned}
 L[k] &= G[k] \Leftrightarrow \text{EXPAND}(G[k + 1]), \\
 G[k + 1] &= \text{REDUCE}(G[k]).
 \end{aligned}$$

Les différences pouvant être positives ou négatives, la valeur zéro est habituellement représentée par le niveau de gris 128. Ceci explique la teinte 'grise' d'une image Laplacienne ainsi que la troncature des différences extrêmes. Notons que chaque niveau $L[k]$ de la pyramide Laplacienne dépend du même niveau $G[k]$ de la pyramide Gaussienne mais également du niveau supérieur $G[k + 1]$. De ce fait, la pyramide Laplacienne possède un niveau de moins que la pyramide Gaussienne.

Plus le niveau de la pyramide est élevé, moins il faut de bits pour coder chaque pixel de L_k , grâce à un codage entropique. Cette méthode est utilisée en compression d'image pour leur transfert progressif (algorithme 2). Cette approche incrémentale permet d'avoir une qualité d'image reçue proportionnelle à la quantité d'information transmise. Le contenu sémantique de l'image est généralement compréhensible dès les premières images. Bien entendu, la méthode nécessite un traitement de la part de l'émetteur et du récepteur. Le gain de temps en transmission est dû au codage

moins coûteux des pixels et à la taille réduite des images.

La pyramide Laplacienne, sous le nom de pyramide dynamique [Den88], est également utilisée dans les séquences d'images pour détecter les mouvements d'une image à l'autre en mesurant des flux optiques : la pyramide est tout d'abord calculée pour deux images successives. La mise en correspondance se fait dans un premier temps à un niveau de faible résolution k , et le champ des vecteurs de déplacements est calculé rapidement par recherche d'une énergie qui doit minimiser la déformation et maximiser la similarité. Ensuite, le champ de vecteurs obtenu est interpolé pour initialiser le niveau précédent $k \leftrightarrow 1$ où le traitement est itéré avec raffinement des vecteurs dont le calcul ne nécessite cette fois-ci qu'une mise en correspondance avec le voisinage immédiat. Le traitement se poursuit jusqu'à la base de la pyramide.

2.2.4 *Scale-space*

L'idée du *scale-space* introduite et formalisée par Witkin [Wit83] est ensuite développée par Koenderink [Koe84]. Cette théorie utilisée pour des signaux continus à une dimension est formulée dans le cas discret et à deux dimensions par Lindeberg [Lin90] pour représenter simultanément tous les niveaux de résolution de l'image. Une étude de ces techniques peut être trouvée dans [Dye87].

L'image à traiter est considérée comme faisant partie d'une famille d'images dérivées (le *scale-space*) ne dépendant toutes que d'un seul paramètre d'échelle t qui décrit le niveau de résolution. Pour une image numérique de niveaux de gris dont la formalisation est classiquement donnée par :

$$\begin{aligned} I : \mathbb{R}^2 &\rightarrow \mathbb{R} \\ I(r) &= I(x, y) = \lambda, \quad r \in \mathbb{R}^2, \lambda \in \mathbb{R} \end{aligned}$$

La fonction à définir est K :

$$\begin{aligned} K : \mathbb{R}^3 &\rightarrow \mathbb{R} \text{ telle que} \\ K(r) &= K(x, y, t) = \zeta, \quad r \in \mathbb{R}^3, \zeta \in \mathbb{R} \end{aligned}$$

de telle façon que $K(x, y, 0) = I(x, y) \forall x, y$.

En pratique, chaque image du *scale-space* est obtenue par convolution de l'image originale avec un noyau Gaussien de taille croissante et conserve la taille de l'image originale. Chaque image peut également être vue comme une solution de l'équation de diffusion de la chaleur.

La plupart des approches *scale-space* font une détection des contours en cherchant les passages par zéro du Laplacien (points d'inflexion) pour chacune des images obtenues. La détection permet de montrer un avantage certain du filtre Gaussien : comme le paramètre d'échelle augmente, il n'y a jamais apparition de nouveaux zéros du Laplacien. Malheureusement, il faut remarquer que le passage à zéro du Laplacien n'est pas toujours très bien localisé sur les contours.

Cette approche définit une image naturelle comme une superposition hiérarchique d'un petit nombre de niveaux de description. L'illustration couramment utilisée est

celle des arbres : au niveau le plus fin, un arbre est composé de feuilles dont la structure est veinée. Au niveau suivant, chaque feuille est une unique région. Puis on peut considérer le feuillage tout entier et enfin la forêt.

La technique initialement utilisée possède un défaut majeur : plus la résolution est faible et plus il est difficile de localiser précisément les contours des objets. Dans [Per87], le problème est résolu en utilisant une équation de la chaleur anisotrope, en rendant variable le coefficient de conduction. Ce coefficient est non nul à l'intérieur d'une zone homogène et nul sur ses bords. De cette façon, le filtrage passe-bas est effectué dans chaque région et non pas entre les régions dont les bords restent intacts. Bien entendu, pour avoir cette information, les auteurs calculent un estimateur de contour à base de gradient. Ensuite, le coefficient de conduction est une fonction normalisée (ou même binaire) de cet estimateur. Un seuillage de l'image gradient des images produites donne enfin des segmentations 'contour' comportant chacune des niveaux de détails différents. Cette notion de filtrage adaptatif est reprise et simplifiée dans [SM88].

Dans un domaine relativement proche du *scale-space*, notons le filtrage non linéaire obtenu en morphologie mathématique par des ouvertures ou fermetures successives réalisées à partir d'éléments structurants de formes diverses [Har87], [Mar89], [Toe74].

2.3 Pyramides géométriques

Nous présentons dans les paragraphes suivants des partitionnements géométriques qui sont utilisés en traitement d'image pour résoudre de façon rapide des problèmes de codage, de compression et de segmentation. La structure de ces partitionnements est liée étroitement à la multirésolution. La même approche par division récursive est utilisée. Seules les techniques diffèrent pour prendre en compte les particularités géométriques des tesselles utilisées.

2.3.1 Constructions *Bottom-up* et *Top-down*

En segmentation par structures pyramidales, les différentes modélisations d'une image induisent différents sens de parcours de la pyramide ainsi que différents types de construction. On distingue les constructions *bottom-up* du bas vers le haut et les constructions *top-down* du haut vers le bas. Certaines techniques alternent les deux sens de parcours pour parvenir à leurs fins.

- La construction *bottom-up* considère l'image comme un ensemble de régions de la taille d'un pixel. Au cours de la construction, le nombre de régions diminue et leur taille moyenne augmente par fusion de régions adjacentes qui partagent des propriétés communes. Le processus est itéré jusqu'à ce qu'aucune fusion ne soit possible.

- La construction *top-down* considère l'image à traiter comme une seule région qui doit être divisée de façon récursive jusqu'à ce que toutes les régions de l'image correspondent à un objet de la scène. Tout au long du traitement, le nombre de régions augmente. Cette structure peut être comparée à une pyramide renversée.

2.3.2 Quadtree

Le principe de représentation par *quadtree* ou arbre quaternaire [Sam80] est lié à la propriété de récursivité du maillage carré. Le *quadtree* est une structure de représentation des images binaires permettant la recherche de composantes connexes. C'est également une structure utilisée pour la segmentation d'images en niveaux de gris [Hor74]. La segmentation s'opère en deux phases, une de division et une autre de fusion (*split and merge*).

Phase de division

L'image à traiter doit être de dimension $N \times N$ ($N = 2^m$). C'est le bloc initial B_0 . Le principe consiste à diviser récursivement tout bloc non homogène selon un prédicat défini. La division d'un bloc B_i de dimension $2^{N-k} \times 2^{N-k}$ donne naissance à quatre sous-blocs² $B_0[k+1]$, $B_1[k+1]$, $B_2[k+1]$, $B_3[k+1]$, de dimension $2^{N-k-1} \times 2^{N-k-1}$. Un exemple montre ce découpage sur une image binaire (figure 7). Après chaque nouvelle division, les attributs de chaque bloc nouvellement créé sont recalculés pour être à nouveau soumis au prédicat d'homogénéité. Présentée de cette façon, la technique du *quadtree* suggère la structure d'une pyramide construite à l'envers (l'apex en bas et la base en haut). En effet, le partitionnement initial est représenté par une unique région (l'image entière) et le nombre de régions (ou blocs) augmente lors de la construction de la structure.

Cette description présente le *quadtree* comme un procédé de découpage récursif. Telle quelle, sa mise en œuvre serait très coûteuse en temps de calcul, puisque le test d'homogénéité d'un bloc nécessite la consultation de tous les pixels qui le constituent. Le *quadtree* serait donc construit à partir de la racine, mais chaque pixel serait consulté un nombre de fois égal à sa profondeur finale dans l'arbre (figure 7.c).

Pour ne pas multiplier les tests sur un même pixel, une approche consiste à ne créer un sommet du *quadtree* que lorsqu'il doit effectivement exister dans la structure. Les pixels sont parcourus dans un ordre particulier. Les ordres de Peano-Hilbert et de Morton [Cha91] permettent de parcourir l'image de manière récursive en formant des carrés qui s'assemblent progressivement par quatre, comme le font les blocs d'un *quadtree*.

La résolution finale de la pyramide *quadtree* est fortement liée au seuil fixant la limite du critère d'homogénéité. Elle est également liée à la taille minimale des

²La numérotation des blocs est la suivante :

0	1
2	3

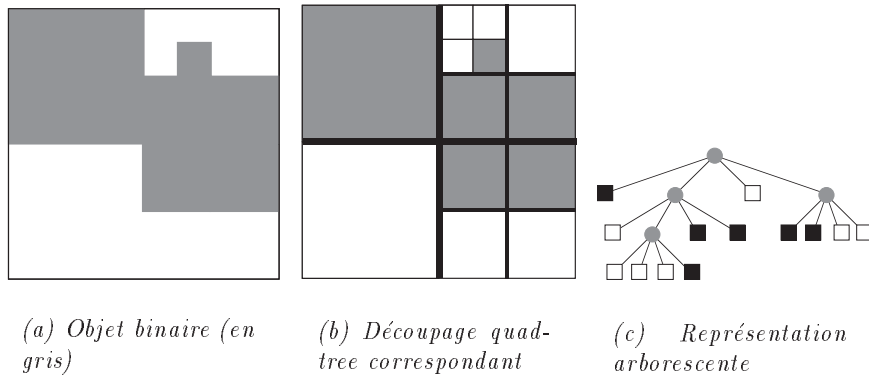


Fig. 7 - Principe de découpage récursif du quadtree

blocs en-dessous de laquelle on ne veut plus provoquer de division quel que soit le résultat de l'évaluation du prédicat d'homogénéité. Celui-ci porte généralement sur la variance des niveaux de gris des pixels qui composent le sommet : un bloc n'est plus décomposé lorsque la variance de ses pixels devient inférieure au seuil défini.

Phase de fusion

La phase de fusion est nécessaire pour regrouper les blocs adjacents représentant des régions homogènes selon le prédicat utilisé, mais qui ont été divisées par le partitionnement régulier quaternaire. Cette phase de reconnexion s'apparente à un étiquetage de composantes connexes [Cha91].

En conclusion, on peut noter que la structure pyramidale *quadtree* est totalement guidée par des critères géométriques stricts. Il n'existe pas d'adaptativité morphologique aux formes des objets présents dans l'image (figure 8.a). De plus, la méthode n'est robuste ni en translation, ni en rotation. En revanche, la mise en œuvre informatique est très aisée : la structure élémentaire est le sommet qui comporte quatre pointeurs sur quatre fils ainsi que la valeur du niveau de gris moyen de ce sommet. Les temps de traitements de la phase de division sont faibles par rapport à de nombreuses autres méthodes.

Pyramides quaternaires

La pyramide quaternaire est une approche structurelle³ à mettre en correspondance à la fois avec l'approche orientée signal de la pyramide Gaussienne (§ 2.2.2) et avec le découpage régulier de l'approche *quadtree*.

La structure simple [Kel71] [ST75] est obtenue en représentant de manière récursive 4 pixels voisins appelés *fils* du niveau k par un pixel *père* du niveau $k + 1$

³Il s'agit d'une structure spécialement développée pour les images numériques.

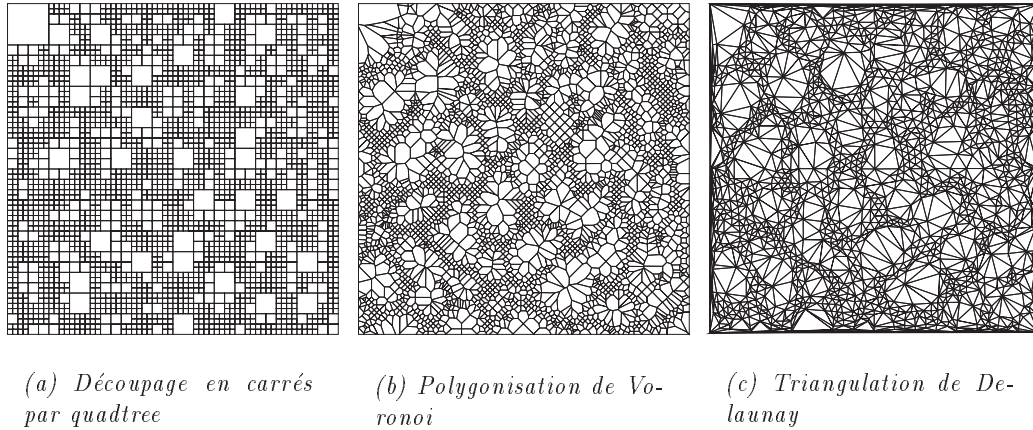


Fig. 8 - Différents partitionnements géométriques de l'image muscle (phase de division uniquement)

(figure 9). L'image doit nécessairement être de taille $N \times N$ (avec $N = 2^n$) et l'apex est atteint en $\log_2(N)$ étapes depuis la base de la pyramide. Cette méthode est très limitée et ne peut pas être utilisée directement en segmentation d'image. Pour ce faire, plusieurs variantes doivent être mises en œuvre (§ 2.3.6).

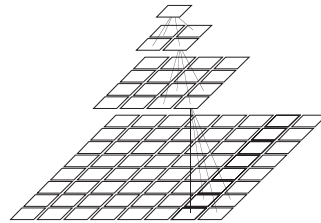


Fig. 9 - Structure d'une pyramide quaternaire. Chaque père possède quatre fils, chaque fils possède un père

La structure quaternaire est utilisée pour effectuer du filtrage (réduction du bruit) tout en préservant les contours. Dans [Par91], les auteurs construisent une pyramide quaternaire dont la valeur des cellules des premiers niveaux sert à mettre à jour l'image originale. Cette mise à jour est faite en fonction du résultat de la comparaison de la variance locale des niveaux de gris composant chaque cellule avec l'estimation de la variance globale du bruit. Près des contours, la valeur des pixels est obtenue par un algorithme de moindres carrés adaptatif. L'élimination de l'artefact classique généré par ce genre de traitement (i.e. l'apparition de structures carrées) est faite

par un moyennage, pour chaque pixel, des résultats obtenus sur l'image translatée [She91].

2.3.3 Polygonisation de Voronoï

La structure de Voronoï est une partition naturelle de l'espace qu'on retrouve dans la nature : alvéoles d'abeilles, bulles de savon [Ste78], molécules chimiques. Cette structure a tout d'abord été utilisée pour la modélisation de l'organisation de populations (astronomie, biologie) puis pour résoudre des problèmes complexes de partitionnement posés par l'urbanisation [Oka92]. Ses propriétés mathématiques sont très intéressantes et permettent une exploitation informatique puissante notamment en analyse d'image.

Diagramme de Voronoï

Un diagramme de Voronoï est un ensemble connexe de polygones de Voronoï. Les définitions suivantes permettent de comprendre la manière dont il est construit :

Définition 2 Soit $M = M_1, M_2, \dots, M_n$ un ensemble de n points du plan euclidien \mathbb{R}^2 . On appelle **polygone de Voronoï** $P(M_i)$ associé au point M_i , l'ensemble des points du plan plus proches de M_i que des autres points de M .

Définition 3 On appelle **diagramme de Voronoï** ponctuel de l'ensemble M , le graphe planaire formé par les frontières des polygones (figure 10.b). Les éléments de M sont appelés *germes* (figure 10.a).

Le diagramme de Voronoï vérifie un certain nombre de propriétés importantes pour sa construction et son exploitation, dont les suivantes :

1. Une arête de Voronoï est un segment de la médiatrice séparant deux germes de l'ensemble M .
2. $P(M_i)$ est un polygone convexe contenant M_i . Les frontières entre 2 régions ont au plus une arête commune.

La structure de données utilisée peut être la suivante (figure 11). A chaque sommet de Voronoï s , est associé :

- les 3 sommets de Voronoï voisins ;
- les 3 sommets de Delaunay créateurs de s .

Cette structure est efficace puisque pour chaque sommet la quantité d'informations est bornée. De plus, une liste associe chacun des germes à un des sommets de son polygone de Voronoï, permettant ainsi d'avoir à la fois le diagramme de Voronoï et son dual, le graphe de Delaunay (figure 10.c).

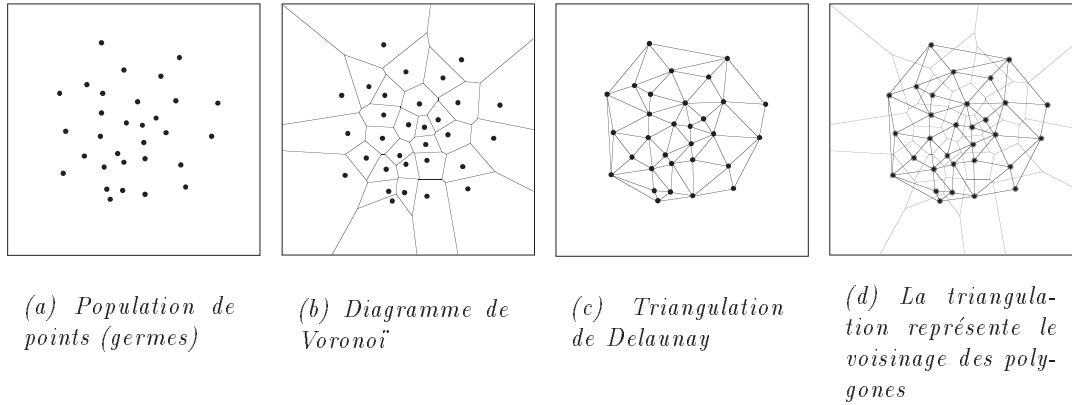


Fig. 10 - Partitionnements de Voronoï et de Delaunay

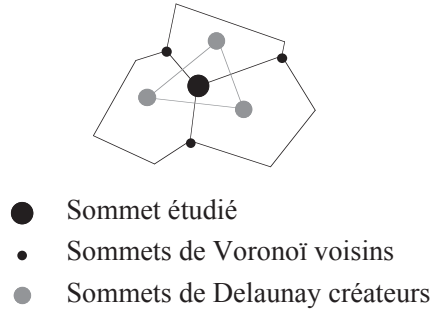


Fig. 11 - Structure de données de Voronoï

Néanmoins, l'insertion d'un germe provoque systématiquement une modification locale de la géométrie des polygones existants. L'algorithme de Bowyer [Bow81] effectue les mises à jour du diagramme de façon incrémentale. Il est utilisé lorsqu'un polygone ne satisfait pas le critère d'homogénéité H choisi. Dans ce cas, un germe est inséré au centre de chacune des arêtes délimitant le polygone non homogène. Lorsqu'un germe est inséré dans un diagramme de Voronoï, un sommet au moins doit disparaître et plusieurs doivent être créés.

Diagramme de Voronoï en segmentation

Le partitionnement de Voronoï est utilisé depuis quelques années en segmentation d'image [Mel92] (figure 8.b). La méthode pyramidale de segmentation [Cha91] est présentée dans l'algorithme 3.

Un des problèmes majeurs consiste à positionner les germes le mieux possible, de façon à ce que les arêtes des polygones épousent au maximum les contours des

Algorithme 3 (*Segmentation par Voronoï*)

```

début
  Générer un petit nombre de germes  $G_i$  sur l'image;
  Construire les polygones  $P_i$  associés;
  Calculer les attributs  $A_i$  de chaque  $P_i$ ;
  Tantque (tous les  $A_i$  ne vérifient pas le critère d'homogénéité  $H$ ) faire
    début
      Pour (chaque  $P_i$ ) faire
        début
          si ( $A_i$  ne vérifie pas  $H$ ) faire
            début
              Insérer un germe  $G_j$  dans  $P_i$  ;
              Mettre à jour localement la structure;
            fin;
          fin;
        fin;
      fin;
    fin.

```

objets [Mel91]. Cette approche peut être utilisée de la même façon en dimension 3 [Ber94].

2.3.4 Triangulation de Delaunay

La structure de Delaunay est duale de la structure de Voronoï. Néanmoins, ses contraintes géométriques sont plus fortes car les régions de base sont des triangles et non plus des polygones convexes (figure 10.c). Les arêtes de Delaunay représentent les relations d'adjacence entre les polygones de Voronoï (figure 10.d).

L'algorithme pyramidal permettant la construction des triangulations de Delaunay successives [Buc88] est identique à celui utilisé pour la polygonisation de Voronoï (§ 2.3.3). Cette structure représente une étape intermédiaire (en ce qui concerne des critères de contraintes géométriques) entre la structure très stricte du *quadtrees*⁴ et la structure pseudo-adaptative des polygones de Voronoï : elle est moins stricte que le partitionnement *quadtrees* car la longueur des arêtes de chaque triangle est variable. En revanche, elle est plus contraignante que la structure de Voronoï car le nombre de côtés de chaque région est constant. La segmentation est donc encore largement guidée par ce facteur géométrique (figure 8.c).

Lorsque le critère d'homogénéité H n'est pas respecté pour un triangle, un germe est inséré à son barycentre. L'algorithme décide alors la façon d'effectuer localement le nouveau partitionnement.

⁴Un pavage triangulaire est plus adaptatif qu'un pavage carré en théorie mais pas pour le système visuel

Fig. 12 - Les 3 niveaux d'une pyramide hexagonale à maillage triangulaire

[Kro85]⁵ propose une structure avec un taux de réduction de 2 (figure 13). Cette structure pyramidale permet de modéliser à différentes résolutions la topologie des contours des objets à l'aide de pyramides de courbes [Kro87].

Dans [Man94], les auteurs proposent une structure pyramidale qui rappelle la distribution des neurones de transmission directe de la rétine : les niveaux successifs ont la régularité d'une pyramide classique quaternaire mais les liens pères fils sont irréguliers : ils sont d'autant plus nombreux qu'on s'éloigne du centre de l'image. Par conséquent, le nombre de cellules diminue du centre vers les bords de l'image. Les images obtenues ont la particularité de recréer le phénomène de la fovéa, c'est-à-dire de conserver un maximum d'informations au centre de l'image et de résumer la périphérie.

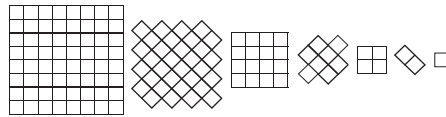


Fig. 13 - Une pyramide avec un taux de réduction de 2

⁵Peu de temps auparavant, [Cro84] présente une pyramide Laplacienne également avec un taux de 2.

2.3.6 Pyramides liées

Les pyramides liées ou pyramides à recouvrements ont été introduites pour la première fois par Burt *et al* [Bur81b]. Elles constituent une première étape vers les structures non régulières. La structure de données de la pyramide liée est caractérisée par les points suivants (figure 14) :

- chaque cellule du niveau k est liée à un bloc de 4×4 cellules du niveau $k+1$;
- au niveau $k+1$, les blocs se recouvrent : chaque bloc de 4×4 cellules recouvre à 25% (c'est-à-dire avec 2×2 cellules) chacun de ses 4 voisins dans le cas de blocs non périphériques ;
- de ce fait, chaque cellule de niveau k est connectée à 4 cellules du niveau $k+1$.

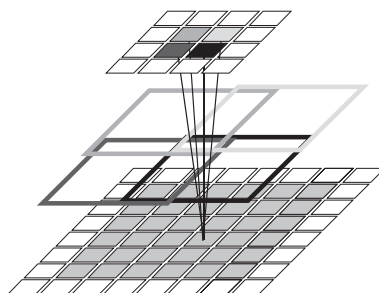


Fig. 14 - Structure d'une pyramide liée. Les 4 pères (colorés) se partagent les fils (grisés) selon 4 groupes (qui se recoupent). Pour plus de clarté, la projection de chaque groupe de fils reprend la couleur de son père. Les 4 fils du centre appartiennent chacun aux 4 pères

Les attributs de chaque père (niveau $k+1$) sont calculés en utilisant les attributs des fils (niveau k). Ces attributs servent à décrire les cellules : niveau de gris moyen, écart-type des niveaux de gris des pixels, ... Lorsque tous les pères ont calculé leurs attributs, chacun des fils choisit, selon un critère de similarité donné, quel est son meilleur père. Si aucun des 4 pères ne convient, le fils devient alors une racine, et son champ récepteur devient une région à part entière. Sinon, le seul lien ascendant conservé est celui qui relie le fils au père choisi. De niveau à niveau, ces liens sont conservés et permettent de reconstituer la composante connexe de chacune des racines.

Hong et Rosenfeld [Hon84] remarquent qu'en segmentation d'image, la plupart des méthodes est fondée sur des décisions de choix forcé, c'est-à-dire qu'une décision prise n'est jamais remise en cause et par conséquent est irrémédiable : appartenance d'un pixel à une classe, division d'une région ou non, fusion de deux régions voisines,

etc ... Ces choix forcés amènent, selon les auteurs, à de nombreuses mauvaises décisions prises alors que peu d'informations étaient disponibles. Pour cette raison, ils introduisent la notion de pyramide reliée. Son principe en est simple, le traitement suivant est itéré jusqu'à convergence : lorsque chaque fils a choisi un père, ce dernier met à jour à nouveau ses attributs, ne tenant compte cette fois-ci que des fils qui l'ont effectivement choisi. En fonction de ces nouveaux attributs, les fils sont à nouveau sollicités pour faire leur choix, et ainsi de suite. La convergence est atteinte si, lors de deux itérations successives, les liens fils-pères restent inchangés. Ces travaux sont repris et la gestion des liens est améliorée et accélérée dans [Cib84].

Les pyramides liées apportent une réponse partielle aux problèmes inhérents aux pyramides rigides. Néanmoins, le nombre possible d'objets ainsi que leur forme (objets allongés), pour un niveau donné, sont réduits. Le problème de non-invariance aux rotations et translations reste également non résolu [Bis90]. Le problème de connectivité, quant à lui, est partiellement résolu par les approches consensuelles.

La mise à jour des liens n'est pas forcément déterministe : Spann utilise une pyramide liée stochastique pour la détection d'objets sur un fond [Spa91]. La mise à jour des liens entre fils et pères se fait de façon stochastique pour permettre à un processus de recuit simulé de converger vers une énergie minimale.

La détection d'objets compacts peut être faite à l'aide de pyramides reliées par la recherche de racines [Hon84]. Lors de la construction de la pyramide (§ 2.3.6) une région est considérée comme une racine si elle appartient à l'apex (de taille 2×2) de la pyramide, ou si à un étage quelconque, la somme des liens avec ses pères est inférieure à un certain seuil. L'ensemble des feuilles d'une racine, également appelé *champ récepteur*, constitue ainsi une région à part entière. Afin de favoriser la compacité des régions, le poids du lien entre un père et un de ses fils prend en compte non seulement un critère de similarité mais aussi un critère de distance (approximation de la distance euclidienne) des centres du père et du fils.

Peleg montre qu'on peut construire des pyramides dont les niveaux peuvent avoir des tailles arbitraires [Pel86], de telle sorte que la réduction de la résolution ne soit pas systématiquement un facteur de puissance de 2 mais variable. La réduction dépend alors des propriétés locales de l'image. L'échantillonnage est effectué en deux passes : sur les lignes puis sur les colonnes de l'image. Les artéfacts dus au sous-échantillonnage sont minimisés car les pixels fils d'un niveau k ont la possibilité d'apporter la même contribution à leurs pères du niveau $k+1$. Par exemple, pour passer d'une ligne de 4 fils (va_k, vb_k, vc_k, vd_k) à une ligne de 3 pères ($vl_{k+1}, vm_{k+1}, vn_{k+1}$), la contribution de chaque fils est $3/4$ (figure 15). La gestion de recouvrements est également possible. Les auteurs calculent un Laplacien moyen pour les lignes et pour les colonnes. La valeur de ce Laplacien indique globalement le degré de haute fréquence dans les 2 directions principales et permet de définir le taux de réduction pour les lignes et pour les colonnes. En outre, notons que cette technique a l'avantage de s'affranchir des effets de bord.

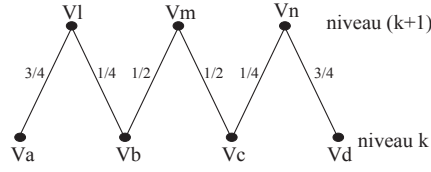


Fig. 15 - Principe d'équicontribution de la pyramide de Peleg

2.4 Pyramides irrégulières

Cette section est consacrée à une introduction rapide aux différents types de pyramides irrégulières, également appelées pyramides de graphes. La pyramide irrégulière doit son nom au type de voisinage particulier des cellules qui la composent : le nombre de voisins de chaque cellule n'est pas fixe, dépend de chaque cellule elle-même et est donc non régulier.

2.4.1 Notions de pyramides de graphes

Dans les pyramides irrégulières, la notion de graphe intervient pour modéliser une relation entre les différentes régions du partitionnement. Chaque région est un sommet du graphe, et la relation représentée par les arêtes est généralement l'adjacence entre les régions. Le graphe d'adjacence est la structure à partir de laquelle vont être extraits des sous-graphes, sur des critères qualitatifs (graphes de similarité, arbres couvrants) et quantitatifs (réduction du graphe) lors de la construction des niveaux successifs de la pyramide.

2.4.2 Pyramides de régions

Pyramide stochastique

Cette structure enlève la contrainte la plus forte liée jusqu'ici aux différentes structures géométriques de base utilisées : carré, triangle, polygone, ... (figure 16). La pyramide est représentée par une suite de graphes d'adjacence (niveaux) où chaque pixel puis chaque région de l'image est un sommet. A chaque niveau, la simplification (contraction) du graphe va provoquer des regroupements entre régions.

La pyramide stochastique peut être utilisée pour l'extraction de composantes connexes d'images binaires [Mon91] : toutes les composantes sont extraites lorsqu'un niveau de la pyramide n'est constitué que de racines. Ce niveau dépend de T_{max} , la taille de la plus grande composante : la hauteur de la pyramide est atteinte en $O(\log_2(T_{max}))$ étapes.

L'utilisation de cette structure pour la segmentation en régions homogènes d'une image en niveaux de gris se réalise de manière hiérarchique [Mon91].

Il est intéressant de remarquer que cette structure est également utilisée comme modèle d'organisation spatiale des plaines alluviales [Chi94] pour effectuer des prédictions d'évolution de milieux géographiques particuliers que sont les réseaux hydrographiques et leur environnement écologique. Le chapitre 4 est consacré à la pyramide irrégulière stochastique.

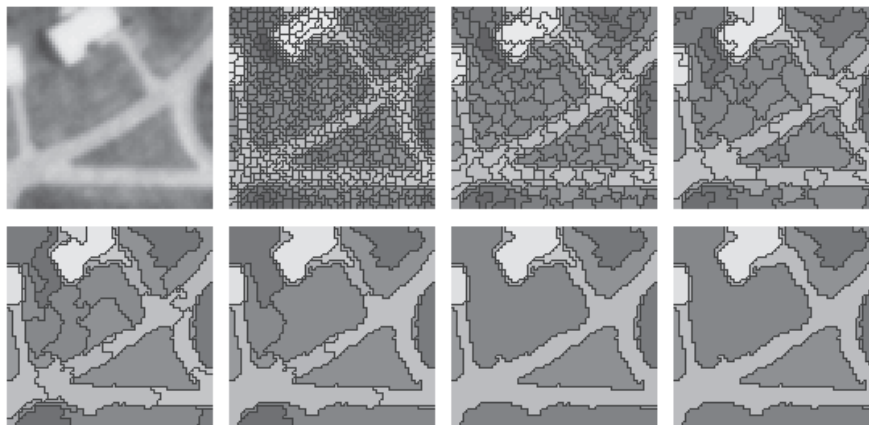


Fig. 16 - Pyramide irrégulière

Pyramide adaptative

La pyramide irrégulière adaptative [Jol92] est une pyramide irrégulière au même titre que la pyramide stochastique. Les deux méthodes diffèrent essentiellement par l'opérateur déterministe qui permet d'effectuer la simplification du graphe (§ 4.2.5).

Dans [Jol92], les auteurs proposent à la suite d'une segmentation de type *bottom up* (§ 2.3.1) avec la pyramide adaptative d'effectuer un traitement *top down* (§ 2.3.1) pour la reconstruction multirésolution des objets : de manière récursive, un objet est décrit à partir de carrés. Ceux-ci sont découpés, leur taille décroît jusqu'à la base de la pyramide où l'objet est un ensemble connexe de pixels. La méthode peut s'apparenter à une sorte de codage *n-tree* (§ 2.3.2) totalement souple.

Pyramide à base arborescente

La notion de pyramide irrégulière a été reprise récemment sous le nom de pyramide souple [Mat93]. Ce type de pyramide désigne une structure irrégulière qui n'est pas forcément initialisée sur un graphe d'adjacence traditionnel. Dans la méthode présentée, un arbre couvrant minimal est utilisé comme graphe initial.

Définition 4 Un **arbre couvrant minimal** (*Minimum Spanning Tree* ou *MST*) est un arbre couvrant tel que la somme du poids des arêtes de cet arbre est la plus faible possible. Cet arbre n'est pas nécessairement unique.

Les pixels de l'image originale constituent d'une manière classique l'ensemble de sommets d'un graphe d'adjacence qui modélise la 4 ou 8 connexité. Chaque arête porte une pondération égale à la différence de niveaux de gris entre les deux sommets qu'elle relie. L'arbre couvrant minimal est calculé sur cet ensemble de sommets. L'idée originale [Suk83] consiste à partitionner l'image en supprimant certaines arêtes de l'arbre couvrant minimal. Dans la pyramide souple, n cellules sont agrégées entre elles en supprimant $n \Leftrightarrow 1$ arêtes. Le phénomène d'agrégation est local et peut donc se produire en parallèle sur tout le graphe grâce à l'algorithme de Sollin.

Pyramide floue

Récemment, [Bra95] introduit la logique floue dans les pyramides irrégulières. Les sommets flous sont les sommets qui retardent le choix de leur père par manque d'information robuste sur leur voisinage. Ils restent attachés à plusieurs pères potentiels jusqu'à ce qu'ils puissent prendre une décision définitive. Ce type d'attachement donne naissance à de fausses adjacences appelées 'arêtes floues'. Cet apport permet aux parties homogènes de l'image de fusionner plus rapidement que les parties d'homogénéité moindre.

2.4.3 Pyramides de contours

La pyramide de contours ne correspond pas à proprement parler à la notion duale de la pyramide de régions. En effet, dans les méthodes existantes, la pyramide de contours ne peut être construite qu'à partir d'une pyramide de régions. Cette dernière est obtenue par une méthode de sous-échantillonnage qui procure des images de taille de plus en plus réduite.

A chaque niveau de la pyramide de régions est appliqué un détecteur de contours. Le principe consiste à relier les contours détectés au niveau k à ceux des niveaux $k \Leftrightarrow 1$ et $k + 1$. Comme la réduction d'un niveau à l'autre de la pyramide de régions se fait par un facteur 4, la correspondance doit se faire entre le père du niveau k et le (ou les) contour(s) porté(s) par la région constituée des 4 fils du niveau $k \Leftrightarrow 1$. Il doit y avoir également correspondance entre les directions des gradients qui ont généré les contours.

2.4.4 Pyramides duales

La pyramide irrégulière pose le problème du voisinage qui peut théoriquement être non borné. Ce point est gênant si on désire construire une telle pyramide sur une machine parallèle, auquel cas on se heurte à un problème de ressources matérielles.

Sur une machine traditionnelle, le problème est cette fois-ci rencontré en programmation où il faut soit gérer un voisinage dynamique, soit utiliser un voisinage

statique de taille assez grande. Dans les deux cas, les temps de traitement en souffrent, et d'un point de vue théorique, l'aspect non-borné n'est pas satisfaisant. Ce problème peut être levé grâce à la notion de graphe dual.

Définition 5 Soit une arête $a = (s_i, s_j) \in A$ reliant deux sommets $s_i, s_j \in S$ du graphe d'adjacence $G(S, A)$. Alors le **graphe dual** de G , \overline{G} , est constitué de faces F et de côtés \overline{A} , tel que $\overline{G} = (F, \overline{A})$. Un côté $\overline{a} \in \overline{A}$ sépare deux faces adjacentes (figure 17).

L'utilisation de cette structure amène deux remarques importantes :

1. Le degré des faces dans un graphe dual n'augmente pas [Kro91b].
2. Le graphe dual possède également la particularité de préserver la topologie des adjacences, comme on peut le voir sur la figure 17 [Wil93].

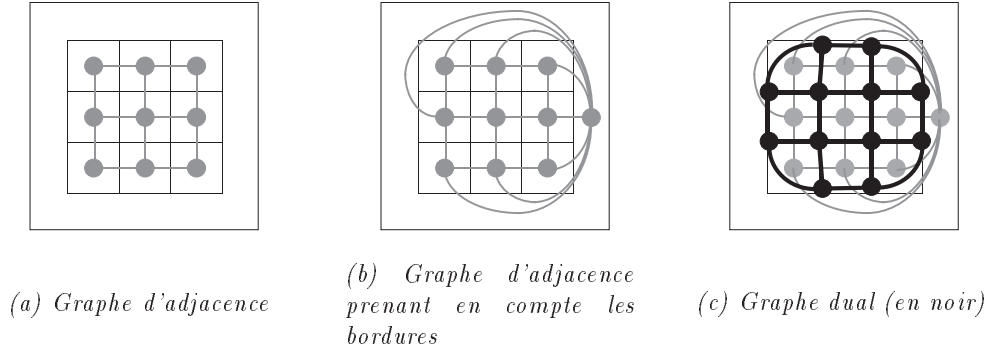


Fig. 17 - Graphe normal vs graphe dual

La construction de la pyramide comporte les mêmes phases que celle de la pyramide irrégulière. Malheureusement, la pyramide duale n'est encore qu'un modèle essentiellement mathématique sans véritable application en imagerie. Les propriétés de cette structure ainsi que sa complexité sont étudiées dans [Wil95].

2.5 Critique des pyramides

Après avoir présenté un certain nombre de techniques pyramidales, il semble opportun de faire le bilan d'une part de l'approche pyramidale en général, et d'autre part des techniques pyramidales de régions par rapport aux autres méthodes de segmentation.

2.5.1 Avantages

En général, les pyramides sont utilisées dans deux environnements principaux : tout d'abord, sur machines parallèles, où elles permettent de résoudre des problèmes particuliers qui nécessitent des traitements en temps réel. Ce premier aspect est d'ordre quantitatif. Ensuite, sur machines séquentielles, elles bénéficient d'approches parallélisables qui permettent des traitements locaux, donc adaptatifs et souvent indépendants entre eux. Ce second aspect est d'ordre qualitatif.

En ce qui concerne le cas particulier des pyramides de graphes, elles possèdent un certain nombre d'avantages par rapport à d'autres méthodes de segmentation. On peut noter principalement, outre les points déjà cités ci-dessus :

- l'image originale n'a pas besoin d'être carrée ni d'avoir une taille $2^m \times 2^m$;
- la robustesse à la translation et à la rotation [Mat93] ;
- la possibilité de représenter des objets de taille et de forme quelconques ;
- la multiplicité des résultats permettant de choisir la segmentation la plus adaptée aux besoins ;
- la faible paramétrisation nécessitée (en général un seul seuil) ;
- la modélisation par graphe fournissant à la fois un cadre mathématique bien formalisé et une grande souplesse d'utilisation.

2.5.2 Inconvénients

Certains inconvénients nous ont paru importants et ont suscité nos travaux :

- les structures proposées ne traitent pas, à l'exception de la pyramide à base arborescente, de l'information de contour ;
- d'une façon plus particulière, ces pyramides génèrent des frontières inexistantes et oublient des discontinuités existantes ;
- la représentation multi-échelle n'est pas à proprement parler une représentation multirésolution, comme dans le domaine *scale-space* ;
- le problème des voisinages non limités est un vaste problème théorique. Cet inconvénient est majeur dès lors qu'on veut construire des pyramides sur des machines parallèles où le nombre de connexions physiques est limité.

2.6 Conclusion

Dans ce chapitre, nous avons traité des modèles de multirésolution en traitement de bas niveau des images. Ces modèles interviennent tout au long de la chaîne de traitement d'image et pas uniquement dans la phase de segmentation. La grande variété et l'imbrication des méthodes rend difficile la caractérisation d'une méthode. On a pu néanmoins distinguer les subtilités entre les notions de structures et de traitements, entre les notions de pyramide, de hiérarchie et de multirésolution.

Par la suite, le chapitre 3 traite exclusivement de technique de segmentation afin de positionner la méthode qui nous intéresse plus particulièrement et qui est traitée en détail dans le chapitre 4.

Chapitre 3

Techniques de Segmentation

La phase de segmentation est sans doute la pierre d'achoppement du traitement d'image. De nombreuses techniques, méthodes, algorithmes peuvent être trouvés dans la littérature. Le niveau de connaissance dans le domaine est tel qu'il est nécessaire à l'heure actuelle de choisir une technique de segmentation adaptée à une application donnée. En effet, chaque technique possède ses avantages, ses inconvénients et ses limites, et les finalités de chacune sont différentes.

Dans ce chapitre, nous présentons certaines de ces techniques en donnant leur fonctionnement général, afin de situer au mieux dans l'état de l'art actuel la méthode sur laquelle nous avons travaillé.

3.1 Problématique

La segmentation d'image joue un rôle prépondérant dans le traitement d'image et dans la vision par ordinateur. Cette étape se situe entre d'une part l'acquisition de l'image et son amélioration et d'autre part sa description et la prise de décision finale conséquente. Elle doit réaliser la tâche difficile d'extraire à partir d'une image numérique des informations 'utiles' qui pourront permettre *in fine* de localiser et de délimiter les entités présentes dans l'image.

3.1.1 Segmentations bas et haut niveau

Le traitement de bas niveau du système visuel est capital dans le processus de la vision. En effet, depuis près d'un demi-siècle, nous savons que le système visuel de la grenouille comporte des couches de neurones spécialisées chacune dans le traitement des informations de contraste, de taille, de forme et de mouvement [Let59].

Plus tard, dès les années 70, les chercheurs estiment que la première étape du traitement visuel chez l'homme et l'animal se fait de façon réflexe : les entités, les formes et les contours émergent automatiquement, de façon inconsciente et indépendamment d'une connaissance *a priori* de l'environnement.

Récemment, Pavlidis argumente qu'en segmentation d'image, la couleur et le mouvement ne jouent pas un rôle important et que la détection de contour par le système visuel est aisée même pour des sculptures abstraites placées dans le feuillage d'un arbre [Pav92].

Quant à nous, partant de ces certitudes, nous plaçons nos travaux parmi les techniques de segmentation de bas niveau : nous considérons comme approche de haut niveau toute méthode pouvant tirer partie d'une ou de plusieurs des informations suivantes : connaissance de la scène, indices visuels particuliers, critères géométriques spécifiques, type de dégradation, nombre d'objets, présence d'un fond, etc... (figure 18). Comme attributs de bas niveau, nous utiliserons donc les notions de pixel (niveau de gris et coordonnées spatiales), de proximité et de connexité.

Néanmoins, il est certain que pour des applications spécifiques destinées à traiter des images particulières, les connaissances *a priori* peuvent et doivent être prises en compte pour optimiser les résultats.

Concluons en remarquant qu'il est capital qu'hormis des segmentations de haut niveau (dans lesquelles les aspects heuristique et expertise prédominent), il existe des traitements de bas niveau qui recherchent à émuler la versatilité de notre système visuel.

3.1.2 But recherché

Le terme **segmentation** regroupe une grande variété de définitions en fonction du but recherché par l'utilisateur. Le traitement peut s'effectuer sur des séquences d'images, pour la poursuite d'objet. Il peut s'agir de focaliser une zone d'intérêt, de



(a) Les chromosomes sur le fond sombre se touchent parfois

(b) Coupe de cordon ombilical. Le nombre d'objets est réduit, les structures sont imbriquées, le fond a un problème d'ombrage

(c) Cette image d'un bâtiment du MIT ne comporte que des structures artificielles rectilignes

Fig. 18 - Différentes images où la connaissance a priori peut être capitale dans la phase de segmentation

délimiter précisément le contour d'un nombre connu d'objets, de détecter toutes les entités d'une image, etc.

En fonction du but fixé, diverses stratégies doivent être utilisées. Celles-ci sont regroupées dans deux approches (ou familles) classiques qui consistent à trouver les régions homogènes de l'image¹ ou le bord des objets² (figure 19). Bien que ces deux approches soient duales l'une de l'autre (une région homogène induit un contour qui l'entoure, alors qu'un contour délimite une région homogène), elles sont aussi et surtout complémentaires : excepté pour un nombre réduit d'images (très bien contrastées par exemple), les deux approches ne fournissent pas les mêmes résultats, mais des résultats qui peuvent être utilisés conjointement.

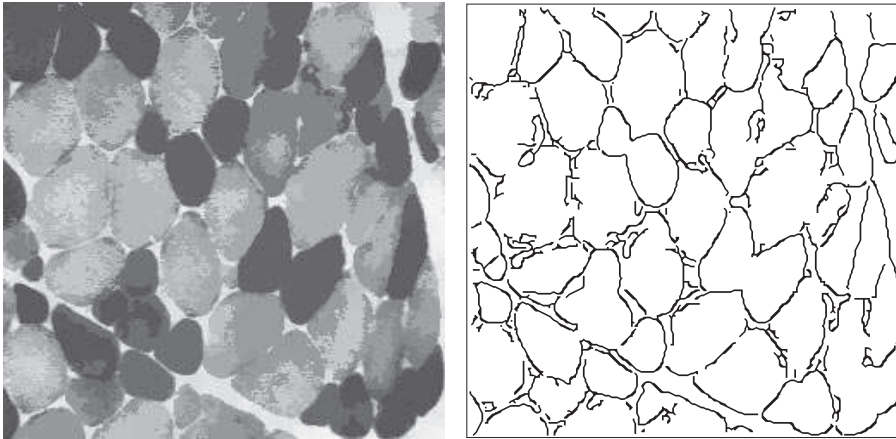
Sur la base de ces deux approches, de nombreuses méthodes ont été développées ces dernières décennies. Le lecteur intéressé par la variété des solutions proposées pourra se reporter aux synthèses de Zucker [Zuc76], d'Haralick et Shapiro [Har85], sans oublier une étude assez récente qui offre une bibliographie abondante [Pal93].

3.2 Approches ‘régions’

Les méthodes de segmentation par approche *régions* sont nombreuses. Elles consistent à rechercher des ensembles de pixels qui possèdent des propriétés communes.

¹L'homogénéité est généralement un critère important qui caractérise la surface de chaque entité.

²Les discontinuités correspondent généralement au passage d'une entité à une autre.



(a) Un partitionnement en régions homogènes munies de leur propre niveau de gris comme résultat d'une segmentation par approche régions

(b) Une carte de contours binaires comme résultat d'une segmentation par approche contours

Fig. 19 - Approches 'région' et 'contour' pour segmenter une image

Les parties connexes de ces ensembles constituent les régions. Le choix des propriétés étudiées est déterminé par le (ou les) critère(s) de segmentation. L'homogénéité d'une région peut être définie de plusieurs façons : niveaux de gris, couleur, texture, etc... Par la suite, nous décrivons des méthodes ou nous donnons les bases de certaines familles de techniques. Pour plus de détails sur les techniques de segmentation, le lecteur peut se reporter à l'ouvrage [Coc95a].

3.2.1 Formalisme

Les méthodes de segmentation par régions fondées sur des critères d'homogénéité peuvent être résumées par le formalisme suivant [Zuc76] qui consiste à construire une partition de régions homogènes :

Définition 6 Soit I une image et E une partition de I constituée de sous-ensembles connexes.

$$E = R_1, R_2, \dots, R_n \text{ avec } R_i \neq \emptyset \text{ et } R_i \text{ connexes, } (\forall i = 1 \dots n)$$

$$I = \cup R_i, (\forall i = 1 \dots n)$$

Soit P un prédicat d'homogénéité appliqué sur un ensemble de pixels. Alors E est une **segmentation** de I , selon le prédicat P , ssi :

1. $P(R_i) = \text{vrai}, \forall i = 1 \dots n.$
2. $P(R_i \cup R_j) = \text{faux}, \forall i \neq j$ tels que R_i et R_j sont adjacents.

3.2.2 Classification

Cette approche est souvent utilisée pour isoler un objet du fond ou parfois plusieurs objets d’une scène. Elle consiste à déterminer une classification des pixels dans l’espace des luminances, en utilisant les niveaux de gris présents dans l’image. Suite à la phase de classification, le niveau de gris moyen de chaque classe est affecté à tous les pixels de cette classe et chaque composante connexe d’une même classe constitue un objet (figure 20). La classification des luminances s’effectue généralement à partir de l’histogramme des niveaux de gris présents dans l’image : on effectue une recherche des modes et des vallées de l’histogramme. Ainsi, la partition obtenue correspond à l’ensemble des n intervalles de niveaux de gris formant chacun un pic et limités de chaque côté par un minimum local. La difficulté de cette approche consiste à déterminer le nombre de classes présentes dans l’image ainsi que les niveaux de gris délimitant chaque classe. Un algorithme représentatif de cette approche est celui de Fisher [Fis58]. L’unique paramètre à fournir est le nombre n de classes que l’on veut discerner dans l’image. Après calcul de l’histogramme des niveaux de gris, l’algorithme recherche n classes optimales telles que la variance de l’ensemble des pixels de chaque classe soit minimisée. Cette méthode est très rapide, simple à utiliser, mais ne tient pas compte de l’aspect spatial de l’image (i.e. de la position des pixels dans l’image). Notons que lorsque le nombre de classes est égal à 2, cette méthode correspond à un seuillage simple de l’image.

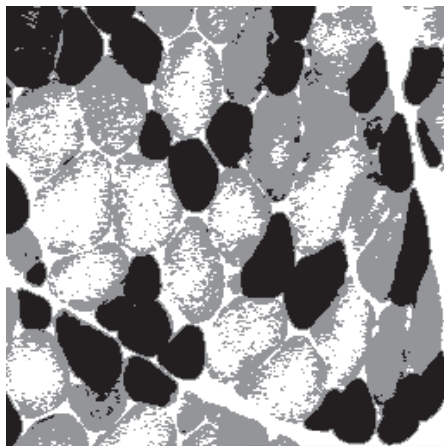


Fig. 20 - Segmentation par classification des pixels (exemple avec 3 classes)

3.2.3 Croissance de région et agrégation

La méthode de croissance de régions est une méthode de fusion. L’image est initialement décomposée en primitives ‘régions’ (une région pouvant être composée d’un seul pixel). Celles-ci sont ensuite regroupées de manière itérative selon un ou

plusieurs critères de similarité, jusqu'à ce qu'il n'y ait plus de fusion possible. On peut distinguer deux sous-méthodes qui consistent soit à procéder itérativement par agrégation de pixels, soit par regroupements itératifs d'ensembles de pixels.

Agrégation de pixels

L'agrégation de pixels est initialisée avec un ensemencement de pixels de départ appelés germes, judicieusement choisis (dans les zones à faible variance par exemple). Chaque germe fusionne avec un premier pixel pour donner une petite région. Le processus est itératif et chaque région croît pixel par pixel. Des heuristiques très simples peuvent être utilisées : un pixel est conquis si la différence entre son niveau de gris et la moyenne des niveaux de gris de la région est faible.

Récemment encore, la méthode a été remise au goût du jour avec un algorithme simple et rapide [Ada94]. Cette technique fonctionne assez bien sur des images peu texturées et sans grands dégradés. Dans les autres cas, elle n'est pas très performante car un pixel ne véhicule que peu d'information.

Croissance de régions

La méthode par regroupements itératifs utilise un procédé relativement similaire en regroupant des ensembles de pixels. Le partitionnement est modélisé par un graphe d'adjacence. Un coût de fusion est associé à chaque couple de régions adjacentes. L'ensemble des régions initiales peut être constitué par l'ensemble des pixels de l'image. Le principe général consiste à fusionner le couple de régions adjacentes dont le coût est minimal sur l'ensemble du graphe et inférieur à un seuil prédéfini. Après chaque fusion, le graphe est mis à jour.

Dans [Gag85], les auteurs utilisent une technique plus affinée en partant d'une partition de petites régions obtenues à la suite d'une phase de division-fusion (§ 2.3.2 et 2.3.2) utilisant un critère très contraignant, et en effectuant plusieurs phases de fusions avec, à chaque fois, un critère de plus en plus tolérant pour mettre à jour le graphe d'adjacence : ces critères portent successivement sur le niveau de gris moyens des régions, sur leur variance, sur le gradient à leur frontière et enfin sur leur surface.

La phase de fusion est une phase critique qui peut être parallélisée de différentes manières pour des coûts de fusion totale non optimaux [Til88], [Bev89], [Cle91], [Cha89].

Comme nous le verrons dans le chapitre 4, la pyramide irrégulière est également une technique de croissance de régions parallélisable.

3.2.4 Morphologie mathématique

La morphologie mathématique est utilisée en traitement d'image non seulement pour partitionner une image en zones homogènes selon certains critères, mais aussi pour isoler directement les objets à étudier. L'outil utilisé est la **ligne de partage des eaux** et une analogie peut être faite avec la croissance de région.

Un ouvrage récent [Sch93] couvrant l’aspect théorique de la morphologie mathématique ainsi que les différentes applications en traitement d’images en niveaux de gris est conseillé au lecteur intéressé par le sujet.

Une image en niveaux de gris peut être considérée comme le relief accidenté d’un sol où la teinte correspond à l’altitude (sombre pour les basses altitudes, claire pour les hautes). Le relief de ce terrain peut être modélisé en $2D$ par des lignes de niveau. Pour connaître ses points les plus élevés, on conserve uniquement les lignes de crêtes. Sur une image en niveau de gris, ces lignes de crêtes peuvent être, par exemple, les maximums locaux du module du gradient dans la direction du gradient.

Soit f la fonction des niveaux de gris de l’image définie sur \mathbb{R}^2 . Pour trouver la ligne de partage des eaux de f , il faut noyer progressivement f (i.e. le relief accidenté) de telle façon que le niveau de l’eau monte à la même vitesse partout (figure 21.a). Ce faisant, l’eau montante forme des lacs qui progressivement peuvent fusionner entre eux. Lorsque deux lacs vont fusionner, il faut construire un barrage à l’endroit de la fusion pour éviter que leurs eaux ne se mélangent (figure 21.b). Le phénomène de déluge doit continuer jusqu’à ce que tout le terrain soit immergé. Seul subsiste l’ensemble des barrages (figure 21.c). Celui-ci (projeté verticalement sur le sol du terrain) forme la ligne de partage des eaux. L’ensemble des lacs (projetés verticalement sur le sol du terrain) forme les bassins versants.

Pour segmenter une image à l’aide de la *lpe*, la fonction f peut être soit l’image originale, soit une image des modules des gradients. Malheureusement, utilisée telle quelle, la *lpe* donne un résultat sur-segmenté dû aux nombreux minimums locaux dont la plupart sont non significatifs. Leur suppression peut être obtenue en contraignant la *lpe* soit par des marqueurs, soit par le contraste :

- Pour construire la *lpe* contrainte par marqueurs, il faut connaître les objets présents dans l’image et associer un marqueur à chacun d’eux. Pour chaque marqueur on calcule un minimum régional en effectuant une érosion géodésique de l’image qui élimine les minimums locaux. On obtient ainsi une *lpe* qui représente un bassin versant par objet et un bassin versant pour le fond. Le problème consiste à localiser les marqueurs.
- La seconde méthode élimine les minimums locaux selon un critère de contraste. Ce critère correspond à une valeur de contraste minimum que l’on oblige entre deux bassins versants voisins. Si le contraste est inférieur, le minimum local lui correspondant est éliminé et les bassins versants sont fusionnés.

L’intérêt de la *lpe* en segmentation d’image réside dans le fait que les contours produits sont fermés. Cette méthode étant très sensible aux minimums locaux (bruit), une phase préalable de lissage est souvent nécessaire.

3.3 Approches ‘contours’

Les méthodes de segmentation par approche ‘contours’ sont sans doute encore plus nombreuses que les méthodes par approche régions. Leur but est de détecter

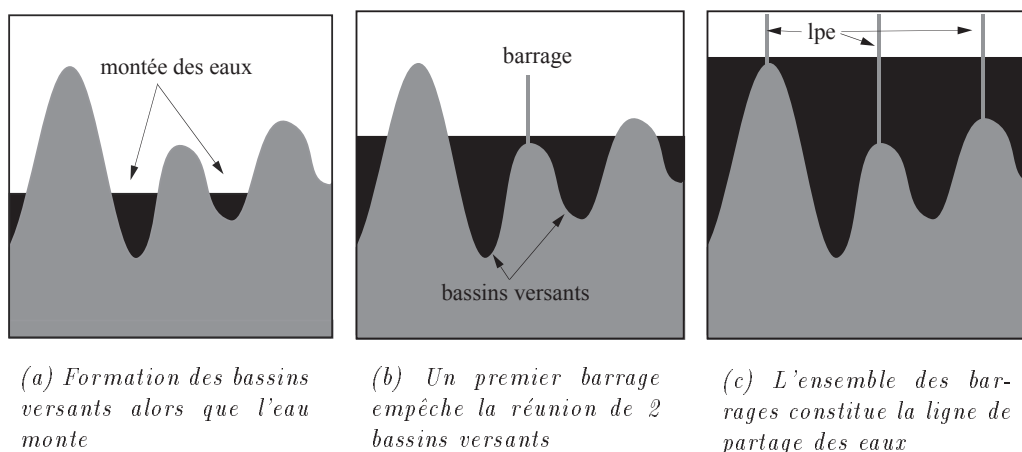


Fig. 21 - Principe d'obtention de la ligne de partage des eaux (lpe) sur une fonction monodimensionnelle

les discontinuités (différences d'intensité) de l'image. En règle générale, une discontinuité correspond à la frontière entre deux objets. Les problèmes rencontrés sont nombreux, et sont dus essentiellement au bruit généré par les capteurs, à l'éclairage et aux phénomènes d'ombrage.

Dans cette section, nous présentons rapidement les principes de différentes méthodes qui ont chacune un intérêt particulier. Le lecteur intéressé par plus de détails peut se reporter à l'étude de Peli [Pel82] qui étudie les performances de certains détecteurs, ainsi qu'au rapport très complet établi par le Groupement de Recherche 134 du CNRS [GRE91].

3.3.1 Approches dérivatives

Les approches dérivatives sont toutes fondées sur l'utilisation des dérivées des fonctions de niveaux de gris. Ce sont les plus immédiates pour détecter et localiser les variations du signal, mais corollairement, elles sont sensibles au bruit. Quelle que soit la complexité de l'algorithme, le but consiste à rechercher soit les extremums locaux de la dérivée première, soit les passages à zéro de la dérivée seconde, soit le passage à zéro du Laplacien (figure 22). En pratique, les phases suivantes sont effectuées :

1. lissage de l'image avec un filtre Gaussien pour éliminer le bruit ;
2. calcul de l'image I_1 de la norme des gradients ;
3. calcul de l'image I_2 de la direction des gradients ;

4. sélection des gradients localement maximaux de I_1 dans la direction donnée par I_2 ;
5. seuillage (simple ou par hystérésis³).

On peut néanmoins faire la distinction entre les opérateurs dérivatifs simples et les opérateurs utilisant l’approche par filtrage optimum.

Opérateurs simples

Les dérivées directionnelles de la fonction image numérique $I(i, j)$ sont approchées par de simples différences. Il en résulte les deux opérateurs :

$$\frac{\delta I}{\delta i} = \Delta_i I = I(i+1, j) - I(i, j)$$

$$\frac{\delta I}{\delta j} = \Delta_j I = I(i, j+1) - I(i, j)$$

La norme du gradient est alors donnée par

$$||\nabla I|| = \sqrt{(\Delta_i I)^2 + (\Delta_j I)^2} \text{ ou bien } ||\nabla I|| = \max(|\Delta_i I|, |\Delta_j I|)$$

Prewitt [Pre70] et Sobel approximent les dérivées directionnelles suivant les axes verticaux et horizontaux à l’aide d’une convolution de l’image avec deux masques V et H de dimension 3×3 chargés de détecter respectivement les gradients verticaux et horizontaux :

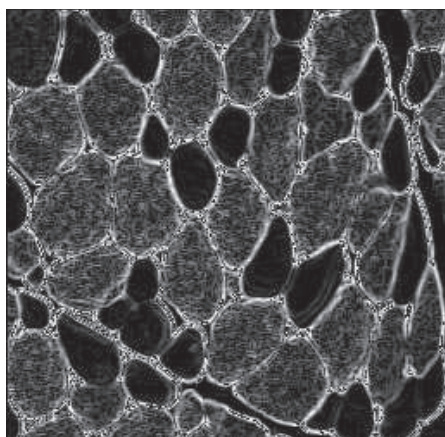
$$V = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix}$$

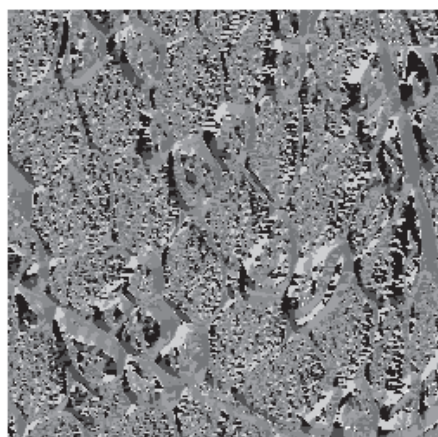
$c = 1$ correspond au masque de Prewitt, $c = 2$ correspond au masque de Sobel. Une valeur de $c > 1$ permet donc non seulement de faire de la différenciation mais également du lissage. Selon le même principe, il existe des masques de dimension plus grande ou des masques multiples. C’est le cas de celui de Kirsh [Kir71] qui propose, à partir d’un unique masque auquel sont appliquées huit rotations de $\pi/4$, de détecter les gradients selon huit directions. La recherche des passages à zéro quant à elle peut être effectuée grâce au Laplacien défini par l’expression classique :

$$\nabla^2 I(i, j) = \frac{\delta^2 I}{\delta i^2} + \frac{\delta^2 I}{\delta j^2}$$

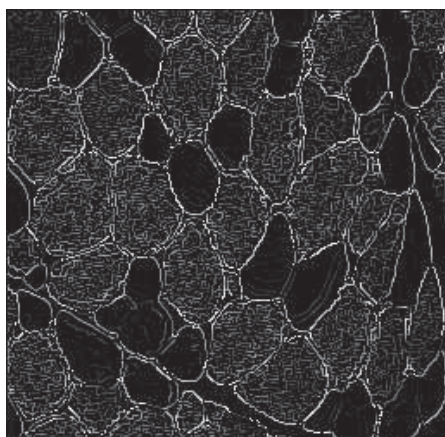
³Le seuillage par hystérésis emploie un seuil bas et un seuil haut. Les maximums locaux dont la valeur est supérieure au seuil bas et qui appartiennent à une composante connexe comprenant au moins une valeur supérieure au seuil haut sont conservés [Can86].



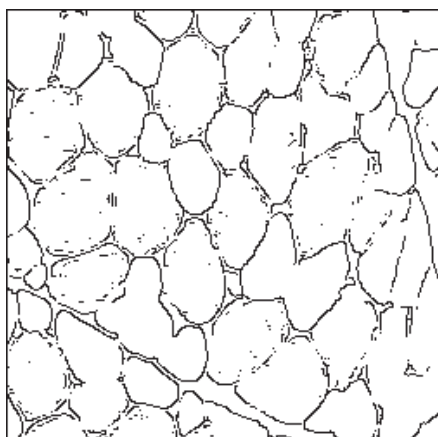
(a) Norme des gradients de l'image 'Muscle' (convolution par les masques de Prewitt)



(b) Direction des gradients



(c) Maximums locaux dans la direction du gradient



(d) Seuillage des maximums locaux

Fig. 22 - Recherche de contours par approche dérivative

En représentation discrète, l'approximation la plus simple du Laplacien est obtenue par la convolution avec un des masques suivants, selon la connexité choisie :

$$L4 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$L8 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

De nombreuses variantes existent sur la base du produit de convolution. Les plus

intéressantes sont celles qui sont adaptatives et qui effectuent un filtrage qui conserve les contours des régions [Per87].

Opérateurs avec filtrage optimal

Dans cette approche, le contour est modélisé par un échelon d’amplitude α noyé dans un bruit blanc. L’image initiale I est d’abord filtrée par un filtre passe-bas (convolution effectuée avec un noyau Gaussien par exemple) pour donner l’image I' . Le filtre optimal est un dérivateur qui permet de détecter des contours en respectant les 3 critères suivants [Can83] :

1. détection : l’opérateur donne une réponse au voisinage d’un contour ;
2. localisation : optimisation de la précision avec laquelle le contour est détecté ;
3. réponse unique : le contour doit provoquer une réponse unique de l’opérateur.

Les opérateurs de Canny [Can86], Deriche [Der87] et Shen et Castan [She86] [Cas89] utilisent tous cette méthode de base fondée sur des calculs différentiels.

3.3.2 Autres approches

Morphologie mathématique

Une fois de plus, la morphologie mathématique [Sch93] offre des outils simples à mettre en œuvre, qui permettent divers traitements intervenant directement sur la forme des objets. On distingue essentiellement deux opérations de base qui sont l’érosion et la dilatation. La ‘quantité’ de matière enlevée (resp. ajoutée) par l’érosion (resp. la dilatation) est en rapport avec la taille et la forme de l’élément structurant (figure 23).

Le gradient morphologique donne une approximation du module du vecteur gradient en un point. Il est noté dans l’espace discret :

$$|\nabla I(i, j)| = \frac{(I \oplus \varepsilon) \ominus (I \ominus \varepsilon)}{2}$$

où \oplus représente l’opération de dilatation et \ominus celle d’érosion. ε est l’élément structurant représenté par le disque unité. Il suffit donc de dilater puis d’éroder l’image en niveaux de gris puisque le gradient s’interprète comme l’étendue, c’est-à-dire la différence entre la valeur supérieure et la valeur inférieure obtenue pour chaque pixel. Ce traitement simple conduit à une sur-segmentation (trop de contours sont détectés). Pour limiter ce phénomène, les images obtenues par les opérateurs de morphologie mathématique sont multipliées avec l’image initiale codée. Cette étape de codage a pour but d’amplifier les niveaux de gris qui correspondent aux contours recherchés. Dans l’image codée, chaque pixel de niveau de gris estimé comme moyen localement, reçoit comme nouvelle valeur le niveau de gris estimé comme maximal localement.

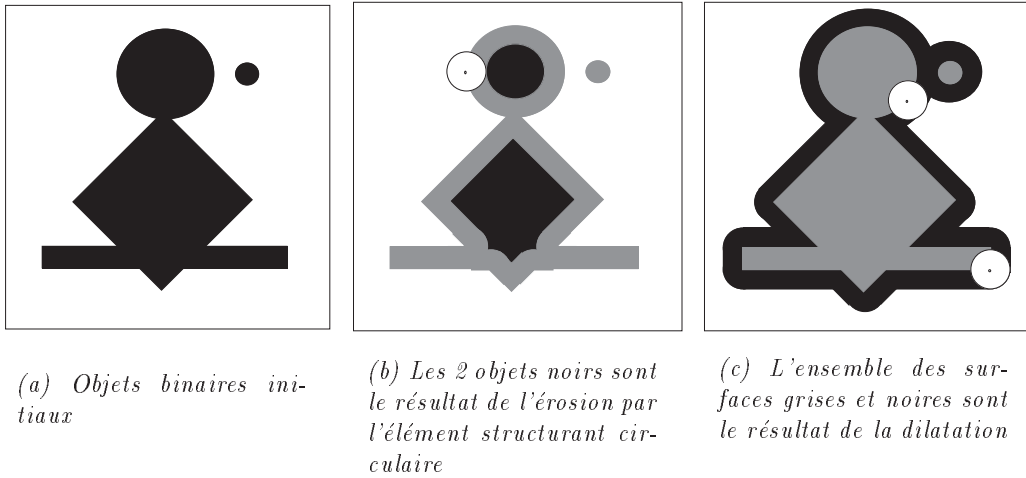


Fig. 23 - Opérations élémentaires de morphologie mathématique

Champs de Markov

Les champs de Markov constituent une approche qui est de plus en plus exploitée en traitement d'image et notamment en détection de contours [Zer90]. Le principe consiste à extraire une fonction caractérisant l'énergie interne du système, et grâce à un certain nombre de paramètres, à le faire évoluer vers l'état qui minimise la fonction d'énergie. Le système, ainsi forcé dans son état de plus basse énergie, est à l'équilibre. Cette évolution vers le minimum d'énergie se fait en étudiant pour chaque pixel son comportement au sein du système et le comportement des pixels voisins. L'interprétation par champ de Markov fournit un modèle mathématique simple permettant de ne prendre en compte que des interactions locales au niveau des pixels. L'inconvénient majeur de cette approche réside dans le coût de calcul très important nécessaire pour des résultats optimaux.

En détection de contours, les techniques utilisant les champs de Markov sont fondées sur l'idée de processus de ligne [S.84]. S. et D. Geman proposent d'introduire un champ booléen dual au processus intensité (les pixels) chargé de représenter explicitement la présence ou l'absence de discontinuité dans le processus intensité et qui implique la création d'un élément de contour. Ces **sites de contours** sont placés entre chaque paire de pixels voisins verticale et horizontale (figure 24.a).

L'ajout de ce treillis de sites virtuels permet alors d'écrire l'énergie totale E de l'image appelée 'énergie de membrane' du système sous la forme :

$$E(I, h, v) = \sum_{i,j} (E_{i,j})$$

L'énergie totale est donc la somme des énergies $E_{i,j}$ de chaque pixel :

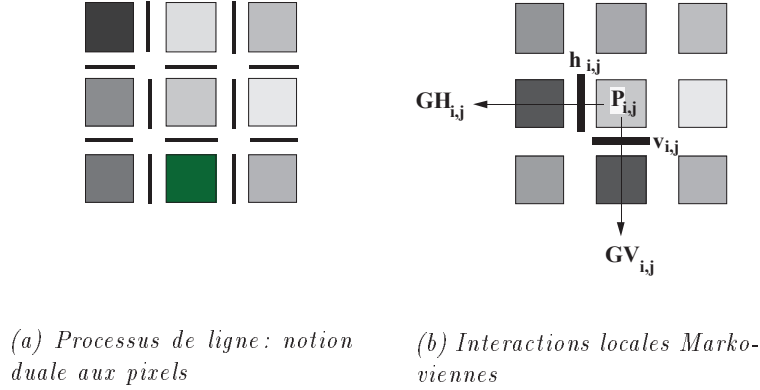


Fig. 24 - Champs de Markov et contours

$$E_{i,j} = E1_{i,j} + E2_{i,j} + E3_{i,j}$$

$E1$ est l'énergie de l'erreur quadratique :

$$E1_{i,j} = (p_{i,j} \Leftrightarrow b_{i,j})^2$$

où $p_{i,j}$ est la valeur du pixel et $b_{i,j}$ la valeur estimée du bruit selon le modèle (on suppose que le bruit est connu pour chaque pixel et est du type blanc Gaussien).

$E2$ est l'énergie de lissage (α est un paramètre) :

$$E2_{i,j} = \alpha(GH_{i,j}^2(1 \Leftrightarrow h_{i,j}) + GV_{i,j}^2(1 \Leftrightarrow v_{i,j}))$$

$GH_{i,j}$ et $GV_{i,j}$ sont respectivement les valeurs des gradients horizontaux et verticaux, $h_{i,j}$ et $v_{i,j}$ sont les valeurs des processus de ligne horizontaux et verticaux. Ces processus de ligne prennent la valeur 1 (présence d'un contour) ou 0 (absence d'un contour) (figure 24.b).

$E3$ est le coût à payer pour un contour (β est un facteur de coût) :

$$E3_{i,j} = \beta(h_{i,j} + v_{i,j})$$

Une fois le modèle d'énergie choisi, celle-ci doit être minimisée en fonction de chacune des valeurs des $h_{i,j}$ et des $v_{i,j}$ afin de déterminer l'état de probabilité maximale du système. Le principal problème réside dans le fait que la fonction d'énergie n'est pas convexe. Il convient donc d'en faire une approximation. Deux algorithmes classiques exécutent cette tâche : le GNC (Graduated Non Convexity) et le Recuit par champ moyen (mean field annealing). Ces algorithmes présentent un intérêt en mathématique probabilistique et ne sont donc pas exposés ici. Ils sont décrits respectivement dans [Bla87] et [Gei91].

3.4 Approches coopératives

Dans le domaine de la coopération régions-contours, nous aborderons tout d'abord les contours actifs qui constituent une approche où régions et contours sont des notions totalement duales.

Ensuite, nous distinguerons deux approches qui sont fondamentalement différentes : la coopération par contrainte et la coopération dynamique. Une étude succincte des méthodes coopératives peut être trouvée dans [Pav90].

Contours actifs

Les contours actifs appelés également *snakes* sont une technique assez récente [Kas88]. Cet ensemble de méthodes consiste à déterminer avec précision le contour d'un objet particulier dans une image. De ce fait, un élément extérieur (l'utilisateur par exemple) doit indiquer de quel objet il s'agit en initialisant le contour actif (figure 25.a). L'initialisation revient à positionner, au voisinage du contour à détecter, une courbe C (ouverte ou fermée) représentée par :

$$C = \{v(s) = (x(s), y(s)); s \in [a, b]\}$$

Par la suite, le traitement va déformer progressivement cette courbe jusqu'à ce qu'elle vienne coïncider avec le contour recherché de l'objet. Le processus de déformation se fait suivant certains critères. Nous présentons ici le critère proposé par [Kas88]. Il consiste à minimiser une énergie qui est celle du phénomène physique de déformation d'une courbe C qui doit coïncider avec le contour d'un objet de l'image. La minimisation est obtenue lorsque les pixels formant la courbe correspondent aux gradients élevés, tout en respectant certaines contraintes en ce qui concerne les propriétés de la courbe (dérivabilité, élasticité, courbure). L'énergie à minimiser sur C est une valeur réelle donnée par :

$$E(C) = E_{image}(C) + E_{interne}(C) + E_{interaction}(C)$$

- $E_{image}(C)$ est l'énergie image Elle modélise les caractéristiques à rechercher dans l'image, c'est-à-dire les forts gradients. Elle peut s'écrire simplement :

$$E_{image}(C) = \Leftrightarrow \int_a^b \|\nabla I(v(s))\|^2 ds$$

Lorsque la norme du gradient de l'image est maximisée, le signe moins de cette énergie participe à la minimisation de l'énergie totale.

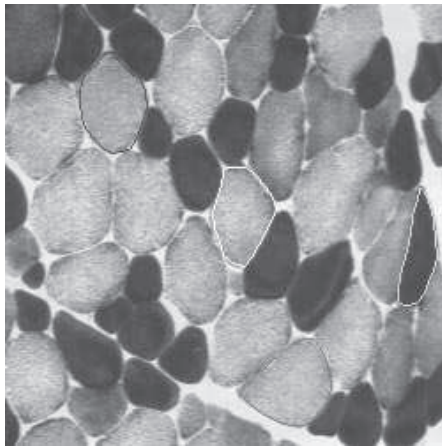
- $E_{interne}(C)$ est un terme régularisant qui permet de rechercher une solution donnant des contours assez réguliers. Ce terme réduit les oscillations dues au bruit présent dans l'image. L'énergie choisie par Kass est la suivante :

$$E_{interne}(C) = \int_a^b \alpha(s) \left\| \frac{\delta v(s)}{\delta s} \right\|^2 + \beta(s) \left\| \frac{\delta^2 v(s)}{\delta s^2} \right\|^2 ds$$

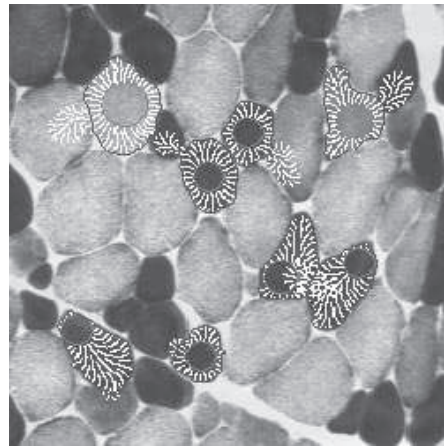
Le premier terme est équivalent à l'équation du déplacement d'une membrane. Le second terme correspond à la vibration d'une plaque mince lors de petits déplacements. Le minimalisation de ces deux termes permet d'obtenir une courbe d'aspect lisse et rigide. Les paramètres α et β permettent de régler le degré d'élasticité de la courbe ainsi que sa courbure et sa résistance à la torsion.

- $E_{interaction}(C)$ est une énergie externe qui provient d'une connaissance *a priori* de l'image. C'est un moyen qui permet de guider l'évolution de la courbe dans une direction particulière.

Ce modèle de contour actif possède l'inconvénient majeur d'être très dépendant de la valeur des paramètres et l'utilisateur doit être confirmé pour obtenir des résultats satisfaisants. D'autres modèles ont été par la suite développés pour palier les faiblesses du modèle initial. Dans le modèle de la bulle [Elo94] la courbe est représentée par une suite de sommets discrets. Sa déformation s'obtient par l'expansion du sommet minimisant une fonction de coût prenant en compte la courbure en ce sommet ainsi que l'interaction avec l'image (figure 25.b). La fonction de coût est similaire à l'expression de l'énergie vue ci-dessus. Ce modèle a la particularité intéressante d'interagir localement avec l'image et d'être ainsi plus adaptatif.



(a) Méthode par snake. Exemple où 2 fibres claires et 1 fibre sombre sont détectées



(b) Méthode de la bulle. Les processus d'expansions (en blanc) partent de la bulle initiale pour rechercher les contours (en noir)

Fig. 25 - Segmentation par contours actifs

3.4.1 Coopération par contrainte

Une des méthodes les plus utilisées en coopération de processus consiste en une croissance de régions qui prend en compte des contours qui ont été extraits au préalable dans l'image originale. Cette prise en compte peut être considérée comme un guidage ou comme une contrainte en fonction du rôle plus ou moins important que prennent les contours. Dans cette section, nous présentons quelques variantes de cette technique.

Croissance de région contrainte

Dans [Wro87], les auteurs mettent au point une coopération entre un détecteur de contours et un détecteur de régions de type croissance de région [Gag85]. La carte de contours de l'image est d'abord extraite ; elle est binaire et peut être le résultat de n'importe quel détecteur. Néanmoins, l'opérateur choisi est celui de Deriche [Der87] qui donne des résultats satisfaisants. Pour diminuer les temps de traitements ultérieurs, une pré-segmentation est effectuée par un partitionnement *quadtree* (§2.3.2) permettant d'obtenir facilement et rapidement un graphe d'adjacence de différentes régions carrées homogènes. Chaque sommet de ce graphe est accompagné de ses attributs (surfaces, niveau de gris, etc...). Les relations d'adjacence modélisées par les arêtes du graphe sont ensuite pondérées par la carte des contours. Les attributs utilisés sont la longueur de la frontière et son nombre de pixels de contours (appelés 'points de contraste'). La fusion de deux régions est alors possible si le rapport entre la longueur de la frontière et le nombre de points de contours sur cette frontière est supérieur à un certain seuil. De toutes les fusions possibles, celle qui minimise la différence des variances entre les deux régions est effectuée en premier. A chaque nouvelle fusion le graphe est mis à jour. La méthode est utilisée pour des applications de vision stéréoscopique.

Plus récemment, [Fal94] développe une coopération originale dans le sens où les germes de la croissance de régions ne sont pas choisis arbitrairement mais suite à une recherche itérative de zones homogènes. Les auteurs apparentent leur approche à une coopération région-région et région-contour.

Dans le cas particulier des images de profondeur, une carte des normales aux entités de l'image peut être calculée afin de détecter des discontinuités sur l'orientation des surfaces. Cette étape grossière est intégrée à une (sur-)segmentation par régions. Une croissance de région permet d'agréger des pixels ou des régions selon un critère qui prend en compte les normales situées sur la frontière des régions [Att91].

Division et fusion contrainte

Pavlidis et Liow proposent une approche quelque peu différente : ils combinent un algorithme de division-fusion avec un processus de détection de contours [Pav90]. La méthode procède en 3 phases :

1. Division-fusion par *quadtree* avec un seuil fournissant une sur-segmentation.

2. Elimination des frontières ; chaque frontière (limite entre 2 régions adjacentes) est conservée ou enlevée en fonction de son contraste, ses changements de direction et sa longueur.
3. Modification des contours. Les frontières conservées lors de la phase précédente sont des contours. La maximisation locale d'une fonction permet d'optimiser leur position à l'aide d'une approche par contour actif discret (snake).

3.4.2 Coopération dynamique

Dans la plupart des méthodes coopératives existantes, le niveau d'intervention des régions est différent de celui des contours. En effet, la notion de coopération dynamique n'intervient pas. Nous entendons par coopération dynamique :

1. d'une part le calcul dynamique des régions ou des contours, en fonction des attributs ponctuellement disponibles ;
2. d'autre part la remise en question bilatérale entre régions et contours, c'est-à-dire la notion de contraintes partagées.

Des techniques fonctionnant selon ces modes ne peuvent qu'être assimilées à des systèmes experts, propres à gérer la distribution des tâches et la diffusion de l'information. Nous illustrons ci-dessous avec deux exemples assez représentatifs.

Coopération semi-dynamique

Dans [Naz84], les auteurs proposent un système expert pour la segmentation de bas niveau. Cette coopération n'est pas totalement dynamique. En effet, une première étape fait une pré-segmentation en régions (chacune étant munie de plusieurs attributs) et calcule une carte de contours. Cette étape détecte également des zones d'intérêt (qui possèdent des attributs identiques) qui vont être tour à tour traitées pour effectuer la segmentation.

Dans un second temps, un certain nombre de règles, qui prennent en compte localement la connaissance que l'on a des régions et des contours (par le biais de leurs attributs), permettent de fusionner des régions, de joindre, de prolonger ou de supprimer des contours. Les zones d'intérêt sont mises à jour dynamiquement.

Il faut noter que, bien que la carte de contours soit mise à jour pendant la segmentation, les résultats restent fortement dépendants de leur détection initiale.

Plus récemment, [Bau92] développe un système expert multi-agents coopératif pour le traitement d'images en cytologie.

Coopération opportuniste

En coopération région-contours, [Sal93] oriente ses recherches sur l'optimisation de l'utilisation des informations plutôt que sur un modèle mathématique. Cette approche nous paraît très intéressante puisque qu'elle se fonde sur l'expertise humaine

pour détecter des indices visuels pertinents pour la construction de régions ou de contours (figure 26). Cette approche consiste, à partir de quelques germes de l'image, à créer des processus 'région' qui croissent et qui font appel, lorsque localement la décision de fusion ne peut être prise, à des processus 'contour'. Toute décision momentanément impossible est repoussée en attendant l'émergence d'informations complémentaires.

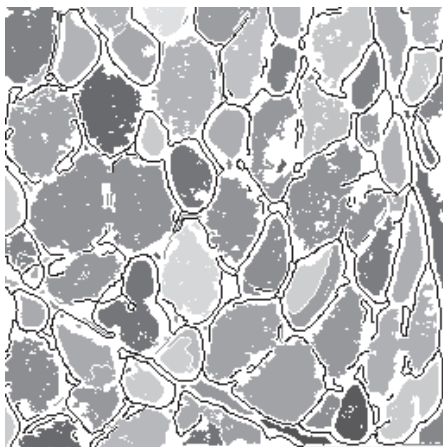


Fig. 26 - Segmentation par coopération dynamique de l'image 'muscle'. Le résultat est un ensemble de régions, de contours et de zones indéterminées (en blanc)

3.5 Conclusion

L'ensemble des chercheurs dans le domaine de la segmentation s'accordent pour affirmer que la coopération des approches 'régions' et 'contours' apporte une meilleure prise en compte des caractéristiques de l'image et donc une meilleure segmentation. Nous croyons, nous aussi, à ce vieux principe général d'intégration qui a maintes fois fait ses preuves, et qui veut que le fruit de l'union de plusieurs méthodes soit supérieur à la somme des fruits de chacune des méthodes. Dans cette optique, nous avons pensé qu'une des façons d'améliorer les performances obtenues avec la pyramide irrégulière était de lui adjoindre une composante 'contours', d'autant plus que cette approche n'avait pas encore été investiguée dans le domaine des pyramides irrégulières⁴. Le chapitre 5 est consacré à ce travail.

⁴Notons tout de même que l'approche de Mathieu (§2.4.2), qui construit un arbre couvrant minimal sur les différences de niveaux de gris des pixels de l'image, constitue d'une certaine façon une prise en compte de l'information 'contours'.

Chapitre 4

Pyramide irrégulière

La pyramide irrégulière est un empilement d'images dont la résolution décroît de la base vers le sommet. Sa particularité principale réside dans le fait que la forme des régions extraites n'est contrainte par aucun critère géométrique.

Dans ce chapitre, nous énonçons les éléments de théorie des graphes utilisés pour la mise en œuvre de la structure. Nous détaillons phase après phase la construction et l'utilisation de la pyramide irrégulière dans le cadre de la segmentation d'images en niveaux de gris. Nous faisons le point sur les résultats expérimentaux obtenus avec cette approche. De ceux-ci, nous dégageons des axes de recherche.

4.1 Principes et terminologie

La segmentation par approche ‘régions’ et notamment par structure pyramidale irrégulière utilise une représentation par graphe. Celle-ci est totalement adaptée aux relations d’adjacence qui unissent les régions. Elle devient nécessaire pour représenter des relations topologiques entre pixels ou régions : ces relations ne peuvent plus être portées implicitement par une structure qui est régulière. Cette section présente tout d’abord les principes de base sur les pyramides de graphes, puis les éléments de formalisation nécessaires sur les graphes et les sous-graphes.

4.1.1 Principes pyramidaux

La pyramide irrégulière possède des caractéristiques dont certaines se retrouvent dans la plupart des structures pyramidales, mais dont d’autres lui sont propres :

- C’est un empilement de niveaux, chacun d’eux étant la représentation à une certaine résolution de l’image originale.
- La base de la pyramide est l’image originale. Le nombre de régions correspondant est égal au nombre de pixels de l’image.
- Le niveau $k + 1$ de la pyramide est construit à partir du niveau k .
- Chaque région est modélisée par un sommet de graphe d’adjacence.
- Tous les niveaux de la pyramide ont la taille de l’image originale. La diminution de la résolution s’obtient par la réduction du nombre de régions qui composent un niveau.
- Les opérateurs utilisés ne mettent en jeu que des traitements locaux, effectués en parallèle sur tout un niveau.
- Une région à un niveau k ($k \neq 0$ et $k \neq \text{apex}$) possède un nombre quelconque (irrégulier) non nul de voisins du même niveau, un parent unique au niveau $k + 1$, et un nombre quelconque non nul d’enfants au niveau $k \Leftrightarrow 1$ (figure 27).

Quelques éclaircissements terminologiques sont nécessaires avant d’aller plus loin : Tout d’abord, la correspondance entre un sommet et sa représentation sur l’image est faite grâce à son champ récepteur.

Définition 7 *Un **champ récepteur** est l’ensemble connexe des pixels de l’image originale qui sont représentés par un sommet de la pyramide, à un niveau donné (figure 28).*

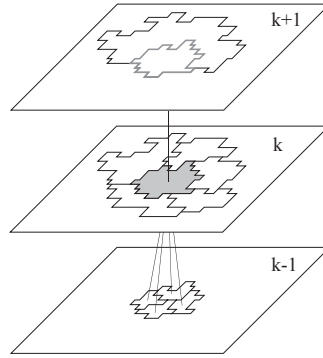


Fig. 27 - Une région de la pyramide irrégulière et les cellules auxquelles elle est reliée (mère, voisines et filles)

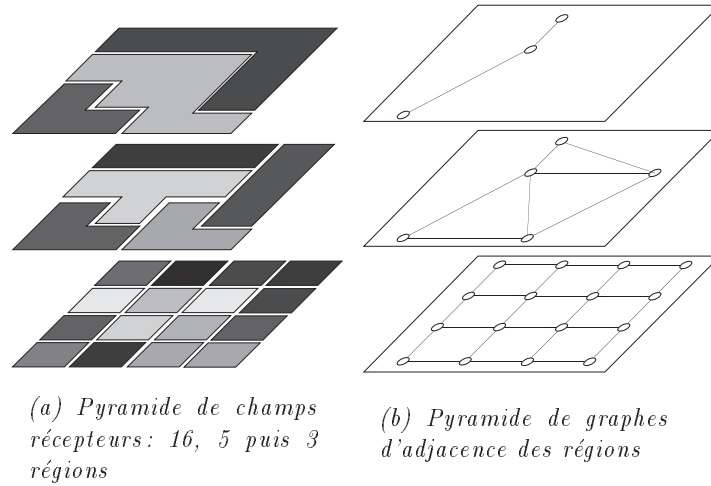


Fig. 28 - Correspondance entre champs récepteurs et graphes. Les graphes servent à modéliser l'information. Les champs récepteurs servent à la représenter

Dans un but de segmentation, tous les pixels d'un même champ récepteur (i.e. d'une même région) ont le même niveau de gris. L'ensemble des champs récepteurs d'un niveau k est un partitionnement de l'image originale.

Définition 8 L'*apex* de la pyramide est le niveau le plus élevé. Ce terme est préféré au terme 'sommet' qui est utilisé pour décrire les éléments d'un graphe.

En segmentation par pyramide irrégulière, l'apex peut contenir un nombre n quelconque ($n \leq N^2$) de régions.

4.1.2 Graphes

L'apport de la théorie des graphes pour les pyramides irrégulières est tout à fait primordial. L'utilisation des propriétés des graphes pour ce type de pyramide est formalisée dans [Bra95]. Cette section apporte néanmoins les éléments nécessaires pour la suite.

Définition 9 Un **graphe** G est un couple (S, A) formé de deux ensembles disjoints $S = s_1, s_2, \dots, s_n (n \geq 1)$ et $A = a_1, a_2, \dots, a_m (m \geq 0)$ tels que, $\forall i, a_i$ est une paire d'éléments de S .

S est appelé ensemble des sommets et A , ensemble des arêtes (figure 29.a).

Définition 10 Un **arc** est un lien orienté¹ qui relie un sommet s_i à un sommet s_j (figure 29.b).

Définition 11 Un **graphe planaire** est un graphe que l'on peut représenter dans un plan sans que des liens (arête ou arc) se recoupent.

Le voisinage de pixels en 4-adjacence donne un graphe planaire, alors qu'en 8-adjacence le graphe est non planaire. Dans la suite du document, on utilisera des graphes non orientés et des graphes orientés :

Définition 12 Un **graphe non orienté** noté $G(S, A)$ est un graphe dont la relation symétrique entre 2 sommets s_i et s_j est indiquée par un lien bidirectionnel appelé arête (figure 29.a).

Par exemple, la relation de voisinage génère un graphe non orienté : si s_i est voisin de s_j alors s_j est voisin de s_i .

Définition 13 Un **graphe orienté** noté $G'(S, A')$ est un graphe dont les liens sont des arcs. (figure 29.b) Ceux-ci servent à modéliser une relation qui n'est pas symétrique.

Par la suite, un graphe non-orienté représentant le niveau k de la pyramide sera noté $G[k] = (S[k], A[k])$, S étant l'ensemble des sommets et A l'ensemble des arêtes. Un graphe orienté représentant le niveau k de la pyramide sera noté $G'[k] = (S[k], A'[k])$, A' étant l'ensemble des arcs.

Enfin, un graphe n'est pas seulement une structure permettant de représenter des relations entre éléments d'un ensemble. Il permet également de quantifier les éléments et les relations utilisés en leur adjoignant des valeurs numériques :

Définition 14 Un graphe $G(S, A)$ est **valué** s'il est muni d'une fonction V :

$$\begin{aligned} V : \quad S &\mapsto \mathbb{R} \\ s_i &\mapsto V(s_i) \end{aligned}$$

A chaque sommet du graphe est associée une valeur.

¹c'est-à-dire unidirectionnel

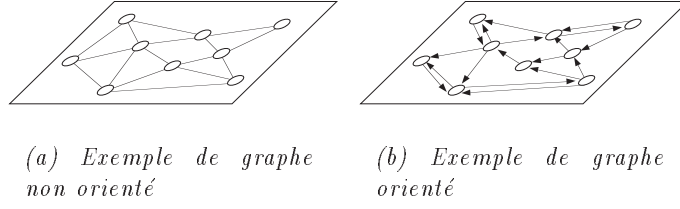


Fig. 29 - Graphes : exemples

Définition 15 Un graphe $G(S, A)$ est **pondéré** s'il est muni d'une fonction P :

$$P : \begin{aligned} A &\mapsto \mathbb{R} \\ (s_i, s_j) &\mapsto P(s_i, s_j) \end{aligned}$$

A chaque arête du graphe est associée une pondération.

4.1.3 Sous-graphes

Les graphes servent à modéliser des systèmes simples ou complexes, notamment des populations d'entités et leurs relations. Cette représentation peut aussi bien rendre compte d'une situation statique que de l'évolution d'un phénomène décrit par une suite de graphes ou de sous-graphes. Le type d'évolution peut être varié : réduction, contraction, etc. Par la suite nous nous intéresserons aux mécanismes de contraction de graphes qu'on obtiendra par la décimation de leurs sommets² pour permettre la diminution progressive du nombre de régions dans le processus de segmentation.

Définition 16 La **contraction** d'un graphe $G(S, A)$ est une transformation qui produit un graphe $G(S \Leftrightarrow U + s, B)$ (avec $U \subset S$) (figure 30). B est tel que :

$$\forall s_i, s_j \in (S \Leftrightarrow U), (s_i, s_j) \in B$$

$$\forall s_i, s_j \in U, (s_i, s_j) \notin B$$

$$\forall s_i \in S \Leftrightarrow U, s_j \in U, (s_i, s_j) \in B$$

Par commodité, les éléments de $(S \Leftrightarrow U)$ sont appelés *survivants* et les éléments de U *non-survivants*.

Définition 17 La **décimation** d'un ensemble de sommets S d'un graphe est une transformation qui produit un ensemble de sommets T tel que $T \subset S$.

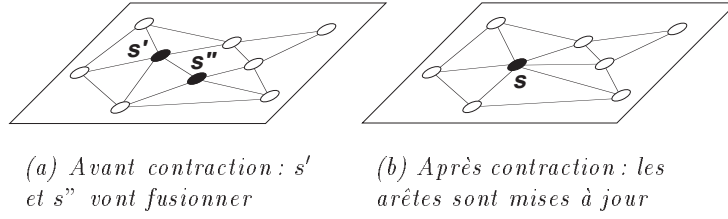


Fig. 30 - Exemple de contraction d'un graphe d'adjacence

Nous verrons que la contraction d'un graphe est directement liée à la notion d'ensemble stable :

Définition 18 Un sous-ensemble X de sommets d'un graphe G est appelé **stable** de G si le sous-graphe G_X engendré par X ne possède pas d'arêtes. Autrement dit, deux sommets d'un stable ne sont pas adjacents (figure 31.e).

Définition 19 Un sous-ensemble X de sommets d'un graphe G est appelé **stable maximal** de G si aucun sommet ne peut lui être rajouté sans perdre la stabilité (figure 31.d).

Toute contraction implique la disparition d'un certain nombre de sommets qui doivent être représentés par les sommets retenus. Ceci pour préserver les relations topologiques entre régions ainsi qu'un partitionnement de l'image. Ce mode de représentation est lié à la notion d'ensemble dominant :

Définition 20 Un sous-ensemble X de sommets d'un graphe G est appelé **ensemble dominant** si tout sommet du sous-graphe $G_{\overline{X}}$ engendré par $G \setminus X$ est adjacent à un sommet de X (figure 31.b).

Définition 21 Un sous-ensemble X de sommets d'un graphe G est appelé **ensemble dominant minimum** s'il est de cardinalité minimale. (figure 31.d).

Définition 22 Un sous-ensemble X de sommets d'un graphe G est appelé **ensemble dominant minimal** si aucun sommet ne peut lui être enlevé sans rompre la dominance. (figure 31.c).

Téorème 1 Un stable maximal est un ensemble dominant minimal (figure 31.d).

Remarque : l'inverse n'est pas toujours vrai (figures 31.c).

²Dans la littérature, *sélection des survivants* est parfois utilisé à la place de *décimation*.

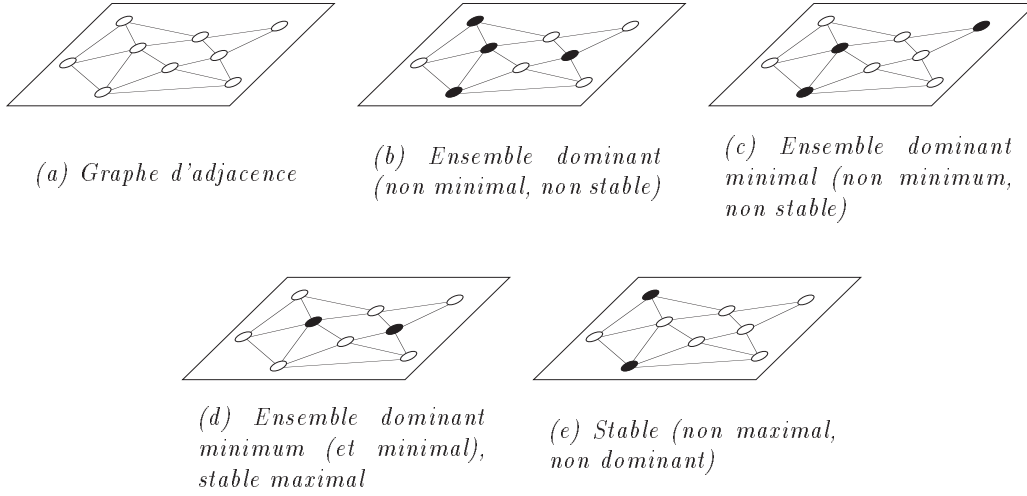


Fig. 31 - Illustration de sous-ensembles de sommets (sommets noirs)

4.2 Construction

Cette section décrit de manière détaillée la construction de la pyramide. Un premier paragraphe permet de montrer de façon intuitive la méthode utilisée. Par la suite, une description des structures de données utilisées ainsi que de l'algorithme général permet de comprendre comment se succèdent les différentes phases de la construction. Enfin, chaque problématique est isolée, et les solutions développées pour la segmentation d'image sont apportées.

4.2.1 De l'image au graphe

La base de la pyramide (niveau 0) est établie à partir des relations de voisinage des pixels de l'image. L'image étant construite sur un maillage carré, il est naturel de relier un pixel à son 4-voisinage ou à son 8-voisinage (figure 32). Le graphe d'adjacence du niveau 0 est donc un graphe planaire ou non, selon que les arêtes correspondent aux relations de 4 ou de 8-voisinage. Ce graphe est non orienté : une arête entre 2 sommets s_i et s_j (au niveau 0, un sommet correspond à un pixel) signifie que s_i est voisin à s_j et réciproquement.

L'image originale peut être vue comme un résultat sur-segmenté : elle comporte autant de régions que de pixels. Le niveau suivant est obtenu par une croissance en parallèle de régions³ dont la conséquence directe est la diminution de leur nombre. Chaque sommet du graphe représente cette fois-ci un ensemble connexe de pixels. Ils sont reliés entre eux par des arêtes qui modélisent leur voisinage. Les régions

³par des traitements appliqués sur le graphe

peuvent avoir une forme quelconque et le nombre de voisins pour chacune d'elles peut varier.

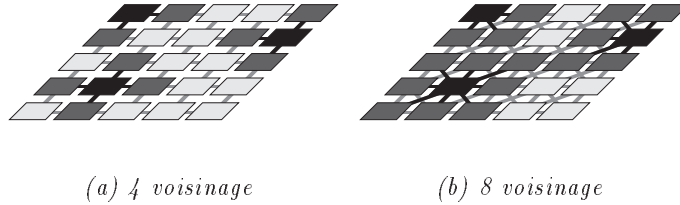


Fig. 32 - Illustration du voisinage (couleur foncée) de pixels (couleur noire) en fonction de leur position dans l'image et de la connectivité utilisée

4.2.2 Structure de données

La structure élémentaire choisie est le sommet de graphe. Chaque sommet représente un pixel à la base de la pyramide et une région sur chacun des niveaux supérieurs par le biais de son champ récepteur.

Chaque niveau est représenté par un graphe et une carte de champs récepteurs correspondante. Le graphe est un vecteur de structures `sommet` qui peuvent être parcourues séquentiellement ou accédées directement par leur indice. La carte des champs récepteurs est une matrice de pixels où chacun d'entre eux pointe sur le sommet qui le représente. La reconstruction des composantes connexes d'une région étiquetée s (i.e. représentée par le sommet s) se fait par récupération de tous les pixels de la carte des champs récepteurs qui pointent sur s .

Pour chaque sommet, les adjacences avec ses voisins sont représentées par un vecteur de pointeurs sur d'autres structures `sommet`. Le lien d'un sommet fils vers un sommet père est fait également à l'aide d'un pointeur. Lorsqu'aucun traitement *top down* n'est effectué⁴, les liens 'sommet père vers sommets fils' ne sont pas nécessaires.

Outre le vecteur de pointeurs vers les sommets voisins et le pointeur vers le sommet père, la structure `sommet` est évaluée par un certain nombre d'attributs descriptifs :

- niveau de gris moyen du champ récepteur ;
- coordonnées du sommet ;
- nombre de voisins ;
- surface du champ récepteur (en pixels) ;

⁴C'est le cas dans la pyramide irrégulière traditionnelle

- seuil local de similarité ;
- écart-type des niveaux de gris du champ récepteur ;
- valeur du tirage aléatoire (pour la pyramide stochastique) ;
- booléen qui indique si le sommet a au moins un voisin survivant ;
- booléen qui indique si le sommet est survivant.

4.2.3 Algorithme général de construction

L'algorithme de construction de la pyramide est de type parallèle : il peut être implanté sur une machine parallèle en utilisant un langage et des techniques appropriés, car les traitements qui sont effectués sont locaux et indépendants entre eux. Le traitement est doublement itératif :

- d'une part, chaque niveau $k + 1$ est construit à partir du niveau k , jusqu'à convergence de la pyramide ;
- d'autre part, l'algorithme de décimation itère un certain nombre de fois sur le graphe du niveau k avant de converger vers un graphe contracté représentant le niveau $k + 1$.

L'algorithme de construction (algorithme 4) est présenté ici de manière succincte, afin de dégager le séquençement des différentes phases. Décrivons sommairement chacune des phases de cet algorithme :

- La procédure **CONSTRUIRE GRAPHE D'ADJACENCE** est chargée d'initialiser la structure pyramidale, c'est-à-dire de créer le niveau 0 à partir de l'image à traiter en utilisant la 4 ou 8-connexité. Le traitement effectué consiste essentiellement à établir les liens du graphe d'adjacence entre les différents sommets⁵ que sont les pixels de l'image : création d'un vecteur de pointeurs pour chaque sommet et mise à jour de ces pointeurs.
- L'algorithme est constitué d'une boucle principale. Elle correspond au traitement effectué pour chaque niveau de la pyramide. Lorsqu'un niveau est construit, le nombre de cellules qui le composent est calculé. La construction de la pyramide s'arrête (i.e. l'apex est atteint) lorsque deux niveaux successifs comportent le même nombre de cellules : le processus de décimation a convergé.

⁵Par la suite, les termes *sommet*, *cellule* et *région* ont la même signification. Leur utilisation dépend essentiellement du contexte : graphe, structure ou image.

Algorithme 4 (*Construction de la pyramide*)

```

début
  niveau <- 0;
  CONSTRUIRE GRAPHE D'ADJACENCE(niveau);
  tantque (NON apex) faire
    début
      CONSTRUIRE GRAPHE DE SIMILARITE(niveau);
      tant que (nombre_survivants[niveau] décroît) faire
        début
          DECIMER GRAPHE DE SIMILARITE(niveau);
        fin;
      ALLOUER NON SURVIVANTS(niveau);
      METTRE A JOUR SURVIVANTS(niveau);
      niveau <- niveau + 1;
      CONSTRUIRE GRAPHE D'ADJACENCE(niveau);
      apex <- nombre_régions[niveau] = nombre_régions[niveau];
    fin;
fin.

```

- La boucle principale de l'algorithme, (i.e. construction d'un niveau), comporte cinq phases successives :
 1. La procédure **CONSTRUIRE GRAPHE DE SIMILARITE** permet de construire le sous-graphe de similarité à partir du graphe d'adjacence après détermination pour chacun des sommets d'un seuil de similarité $T(s)$ qui lui est propre et qui détermine les voisins susceptibles de pouvoir être fusionnés avec lui.
 2. Le traitement qui suit la sélection des voisins similaires est un processus itératif de décimation des sommets du graphe de similarité (procédure **DECIMER GRAPHE DE SIMILARITE**). Après chaque décimation partielle du graphe de niveau k , le nombre de survivants pour le niveau $k + 1$ est calculé. Lorsque ce nombre ne diminue pas au terme de deux itérations successives, le processus s'est stabilisé et la décimation du graphe est terminée.
 3. Après la décimation vient l'allocation des sommets non-survivants (procédure **ALLOUER NON SURVIVANTS**) : chaque sommet non retenu pour former le graphe du niveau supérieur est rattaché à un sommet survivant voisin pour être représenté par lui. Les champs récepteurs associés fusionnent.
 4. Dès lors que les survivants connaissent les sommets non-survivants dont ils ont hérités, leurs attributs peuvent être calculés en prenant en compte leurs propres attributs et ceux de chacun des sommets non-survivants associés (procédure **METTRE A JOUR SURVIVANTS**).

5. La dernière phase (procédure CONSTRUIRE GRAPHE D'ADJACENCE) reconstruit le nouveau voisinage du niveau $k + 1$ à l'aide du graphe d'adjacence du niveau k .

Le processus de construction du niveau $k + 1$ est terminé. Une itération identique peut alors commencer pour construire le niveau $k + 2$ en partant du niveau $k + 1$.

4.2.4 Construction du graphe d'adjacence original

La construction du tout premier graphe d'adjacence est effectuée en début de traitement (procédure CONSTRUIRE GRAPHE D'ADJACENCE). Elle consiste à appliquer l'image originale sur la structure de données choisie et à constituer ainsi la base de la pyramide.

Lors de cette phase, un sommet de graphe est créé pour chaque pixel. Les attributs propres à chaque sommet (§ 4.2.2) sont initialisés de la façon suivante :

- niveau de gris du champ récepteur \leftarrow niveau du gris du pixel ;
- coordonnées du sommet \leftarrow coordonnées du pixel ;
- nombre de voisins $\leftarrow 2, 3, 4$ ou $3, 5, 8$ en fonction de la connexité choisie et de l'emplacement du pixel (figure 32) ;
- surface $\leftarrow 1$ pixel ;
- écart-type $\leftarrow 0$;
- le sommet n'a aucun voisin survivant (*a priori*) ;
- le sommet est non-survivant (*a priori*).

Le seuil local de similarité est initialisé par la suite (§ 4.2.6). La valeur du tirage aléatoire également, dans le cas de la pyramide stochastique. Les pointeurs sur les voisins sont initialisés dans chaque vecteur en prenant en compte la connexité utilisée et l'emplacement du pixel (figure 32).

Chaque niveau suivant est un graphe réduit du graphe du niveau précédent et possède la structure que nous avons vue. Nous allons étudier maintenant les méthodes utilisées pour réduire le graphe niveau après niveau dans le cas de la pyramide stochastique.

4.2.5 Contraction du graphe d'adjacence

Nous avons étudié, dans le chapitre 2, différentes techniques de filtrage et de sous-échantillonnage permettant de réduire la résolution dans une pyramide. Dans cette section, nous étudions les différentes techniques propres à la pyramide irrégulière, qui permettent de décimer un graphe pour diminuer le nombre de régions entre 2 niveaux consécutifs.

Problématique

La réduction du nombre de régions du niveau k au niveau $k + 1$ s'exprime dans notre approche orientée 'graphes' par la réduction du nombre de sommets. Considérons dans un premier temps le cas d'un graphe d'adjacence. Le problème consiste à réduire de façon homogène les sommets, c'est-à-dire à enlever des sommets de façon équitable sur toute la 'surface' du graphe, sans pénaliser ou favoriser une partie de ce graphe. Afin d'effectuer une décimation qui prenne en compte les contraintes dictées par

1. la structure pyramidale,
2. la nécessité de partitionner l'image en vue d'une segmentation,

il convient de fixer les 2 règles capitales suivantes :

Règle 1 *Deux sommets adjacents au niveau k ne peuvent pas survivre tous les deux au niveau $k + 1$.*

Règle 2 *Au niveau k , tout sommet non-survivant doit posséder au moins un sommet de son voisinage qui survit au niveau $k + 1$.*

La règle 1 (facultative) permet une décroissance significative et répartie du nombre de sommets. Cette décroissance correspond bien à une structure pyramidale.

Grâce à la règle 2 (obligatoire), un sommet non-survivant trouve toujours localement un sommet voisin (au moins) survivant. Quel que soit le niveau considéré, l'ensemble des sommets représente donc bien un partitionnement de l'image (figure 34).

La notion de 'stable maximal' vue au § 4.1.2 répond bien au besoin formulé et vérifie les deux règles énoncées : la contraction d'un graphe doit se faire par la recherche d'un stable maximal de ce graphe. La méthode choisie peut être séquentielle ou parallèle, déterministe ou stochastique. Les méthodes parallèles, nous le verrons, obligent des traitements locaux qui sont très bien adaptés à la nature de l'image.

Comme méthode de décimation simple, on peut utiliser un traitement déterministe qui consiste par exemple à sous-échantillonner le graphe de manière régulière (figure 33). Néanmoins cette approche est trop rigide et va à l'encontre du but recherché.

Une autre solution, séquentielle et purement algorithmique, consiste à balayer la suite des sommets du graphe, à sélectionner le premier sommet non marqué, à marquer tous ses voisins, et à rechercher le prochain sommet non marqué pour réitérer le processus jusqu'à la fin de la liste. Après un nombre fini de balayages, la liste ne contient que des sommets marqués ou sélectionnés, et le sous-graphe obtenu est bien un stable maximal (§ 4.1.3).

Enfin, notons qu'un stable maximal est d'autant meilleur qu'il comporte un nombre important de sommets. A partir d'un même graphe, il est possible d'obtenir plusieurs stables maximaux (figure 34).

Nous allons maintenant étudier les algorithmes qui permettent leur extraction.

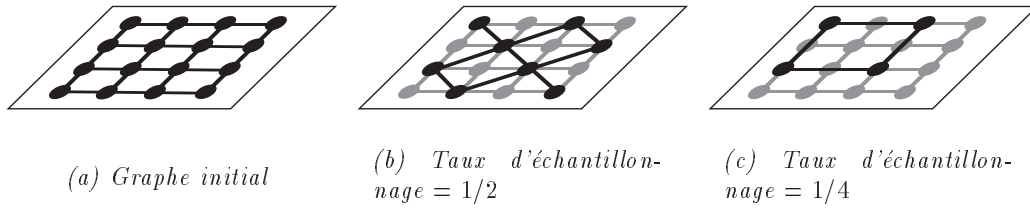


Fig. 33 - Réduction de graphe par sous-échantillonnage. Exemples (les sommets et les arêtes noirs sont conservés)

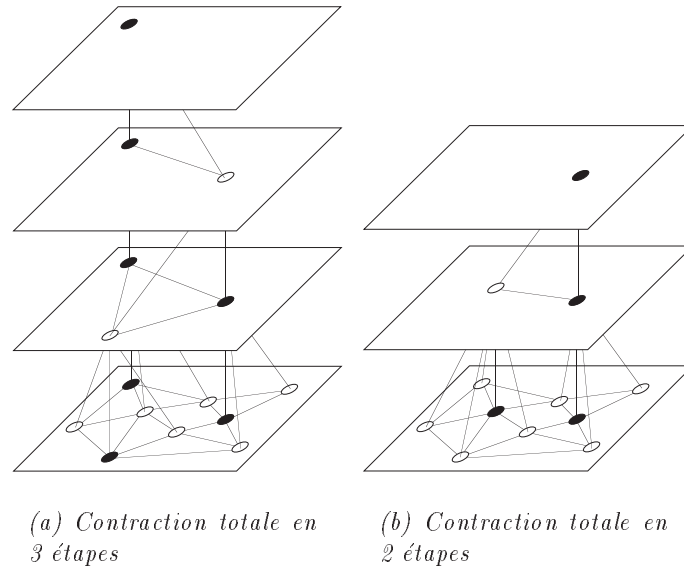


Fig. 34 - Exemple de pyramides de stables maximaux d'un graphe. Les sommets noirs sont survivants, les liens verticaux sont des attachements possibles

Principe général de décimation

Nous nous restreignons ici aux algorithmes parallèles de recherche de stable maximal qui sont fondés sur le même principe (algorithme 5).

Décimation stochastique de Luby

L'algorithme de Luby (algorithme 6) [Lub86] est une application directe du principe général énoncé dans le paragraphe précédent. Soit $score(s)$ la valeur booléenne associée à un sommet s .

Algorithme 5 (*Principe général de décimation*)

```

début
  G[k] = (S[k], A[k]);
  S[k+1] <- ∅;
  S <- S[K];
  tantque (S <> ∅) faire
    début
      Sélectionner I, un stable du sous-graphe induit par S sur G[k];
      Ajouter I à S[k+1];
      Enlever de S, I et tous les sommets adjacents à I sur G[k];
    fin;
fin.

```

Algorithme 6 (*Décimation de Luby*)

```

début
  I <- ∅;
  faire (en parallèle pour tous les sommets s de S[k])
    début
      calculer d(s), le degré du sommet s;
      score(s) <- 1, avec la probabilité P1 = 1/(2d(s)) ou si d(s) = 0;
      score(s) <- 0, avec la probabilité P2 = 1 - P1 et si d(s) <> 1;
      si (score(s) = 1)
        alors ajouter le sommet s à I;
    fin;
  faire (en parallèle pour toutes les arêtes (s_i, s_j) de A[k])
    si (s_i appartient à I et s_j appartient à I)
      alors enlever s_k de I, (k=i, j) avec min d(s_k);
  I est le stable maximal recherché;
fin.

```

La probabilité pour un sommet d'être sélectionné dans l'ensemble dominant intermédiaire est liée à son degré. La procédure de sélection favorise les sommets dont le degré est faible, afin d'augmenter le nombre de sommets retenus, et obtenir ainsi un stable de meilleure qualité. Lorsque deux sommets adjacents sont retenus simultanément, le traitement doit enlever un des deux sommets avant de procéder à l'itération suivante. Pour cette raison, l'algorithme de Luby n'est pas le plus efficace.

Décimation stochastique de Meer et Connelly

Cet algorithme (algorithme 7) [Mee89] procède à une sélection des extremums locaux. Grâce à son approche probabilistique, il évite les problèmes de symétrie. Soit $score(s)$ une valeur réelle aléatoire $\in [0, 1]$ (suivant une distribution uniforme)

associée à un sommet s . Les sommets sont sélectionnés en utilisant un extremum

Algorithme 7 (*Décimation de Meer et Connelly*)

```
début
  I <- ∅;
  faire (en parallèle pour tous les sommets s de S[k])
    début
      tirer aléatoirement score(s);
      si (score(s) > score(s_i) (pour tout i / (s,s_i) appartient à A[k]))
        alors ajouter s à I;
    fin;
  I est le stable maximal recherché;
fin.
```

local de la valeur de $score(s_i)$ (i.e. la valeur maximale dans le sous-ensemble constitué par un sommet et tous ses voisins). Ainsi, deux sommets adjacents ne peuvent jamais être choisis simultanément pour appartenir au stable intermédiaire. Aucun retour arrière n'étant nécessaire, cette méthode est plus rationnelle que la précédente. Elle prend en compte implicitement le degré d'un sommet : plus le degré d'un sommet est élevé, plus la probabilité que son $score(s)$ soit supérieur à tous les tirages de ses voisins est faible. Après implantation et comparaison des deux méthodes, il s'est avéré que les qualités des stables étaient identiques. Néanmoins, la méthode des extremums locaux est plus rapide et a donc été conservée par la suite.

La figure 35 montre sur un support initial de type image (maillage carré et 8-voisinage) les différents niveaux d'une pyramide, construite par recherches successives de stables maximaux avec l'algorithme de Meer et Connelly.

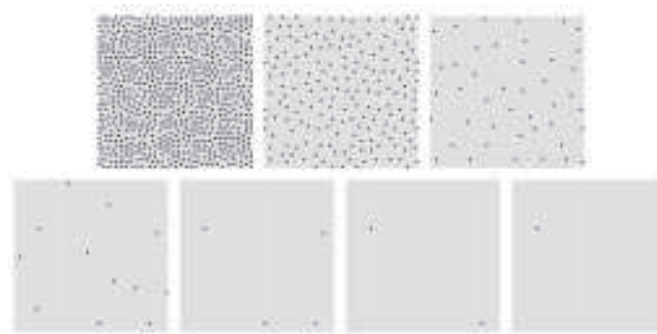


Fig. 35 - Répartition des sommets de stables maximaux obtenus avec l'algorithme de Meer et Connelly. Chaque niveau est un stable maximal du niveau précédent

Décimation adaptative

Dans la pyramide adaptative présentée dans [Jol92], le choix des sommets survivants n'est pas fait de manière stochastique mais par rapport au contenu de l'image. A chaque sommet s est associé un score (appelé **opérateur d'intérêt**) qui est égal à la variance $\sigma(s)$ des pixels du champ récepteur du sommet. A la base de la pyramide, comme chaque sommet est un pixel, $\sigma(s)$ est calculée sur la fenêtre 3×3 représentée par le pixel et son 8-voisinage. Un sommet survit s'il minimise localement l'opérateur d'intérêt, c'est-à-dire s'il est plus uniforme que ses voisins. Le sous-graphe calculé est aussi un stable maximal. Dans le cas où plusieurs sommets voisins minimisent le score, la méthode a néanmoins recours à un tirage aléatoire pour décider du survivant.

En pratique, le cas est rencontré uniquement lors de la décimation de la base de la pyramide ou sur plusieurs niveaux pour des zones ou des images uniformes.

Il est intéressant de noter que puisque la notion de survivance s'effectue sur un critère d'homogénéité (qui est assez proche de la notion de similarité), la recherche du stable maximal est faite directement sur le graphe d'adjacence.

Décimation par réseaux de Hopfield

Un réseau de Hopfield est un réseau de neurones constitué d'unités binaires i (de valeur 0 ou 1) et reliées à toutes les autres unités j par une connexion qui leur permet d'émettre leur état affecté d'un poids ω_{ij} .

Ce réseau peut facilement être interprété comme un graphe d'adjacence valué et pondéré. Hopfield a montré que lorsque certaines conditions sont respectées, le réseau converge toujours vers un état stable. Celui-ci est mesuré par une énergie minimisée lorsque la valeur de chaque sommet est mise à jour par un traitement itératif, de la manière suivante :

$$score(s_i) = \begin{cases} 1 & \text{si } \sum_{(s_j, s_i) \in A} \omega_{s_j s_i} score(s_j) > 0 \\ 0 & \text{sinon} \end{cases}$$

Les pondérations des arcs sont choisies de la façon suivante :

$$\omega_{ij} = \Leftrightarrow f(\Delta_{ij}) \text{ avec } \Delta_{ij} = |g(s_i) \Leftrightarrow g(s_j)|$$

$g(s_i)$ et $g(s_j)$ sont les niveaux de gris de s_i et s_j . f est la fonction définie par :

$$f(\Delta_{ij}) = \begin{cases} 2 & \text{si } (s_i, s_j) \in A \text{ et } \Delta_{ij} \leq T \\ 0 & \text{sinon} \end{cases}$$

T est un seuil. Dans la pratique, sa valeur est comprise entre 7 et 10. L'application à la contraction de graphe dans les pyramides irrégulières est directe [Bis94] [Ber95a]. L'état initial du réseau de Hopfield peut être choisi de manière aléatoire. Les expérimentations montrent que l'algorithme converge dans la plupart des cas en 6 itérations.

Expansion des régions

Nous venons d'étudier le principe de réduction du nombre de sommets dans un graphe d'adjacence. Dans ce paragraphe, nous allons appliquer la recherche de stable maximal au partitionnement de l'image.

Dans le processus de réduction, pour assimiler une région à un sommet, il faut respecter l'aspect topologique. Pour cela, le champ récepteur associé à un sommet survivant croît en annexant les champs récepteurs de sommets voisins non-survivants.

Dans notre exemple (figure 36) la partition originale est composée de 4096 régions représentant les 4096 pixels de l'image. A chaque région est affecté un sommet du graphe d'adjacence. Au cours des niveaux successifs, le nombre de sommets diminue, diminuant ainsi le nombre de régions.

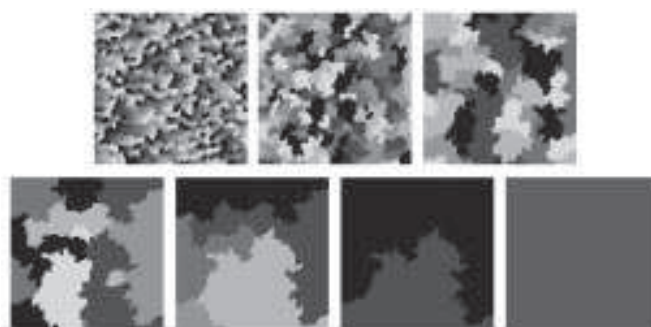


Fig. 36 - *Pyramide de champs récepteurs. La coloration des régions n'a pas de signification intrinsèque et permet uniquement de distinguer leurs frontières*

4.2.6 Construction du graphe de similarité

Nous avons vu comment gérer la dualité *sommet* - *région* dans la construction pyramidale. Il faut maintenant adapter ce traitement au contenu de l'image, c'est-à-dire prendre en compte non seulement la notion d'adjacence entre régions, mais également leur similarité.

Graphe de similarité non orienté

Le processus de segmentation doit permettre de respecter l'homogénéité des niveaux de gris dans chaque région. Cette homogénéité passe par la similarité des régions qui fusionnent.

Pour prendre en compte les caractéristiques de l'image, chaque stable maximal est extrait à partir d'un graphes de similarité et non plus du graphe d'adjacence. En d'autres termes, les règles 1 et 2, définies en § 4.2.5 doivent être respectées maintenant sur le graphe de similarité (figure 37). Néanmoins, le graphe d'adjacence

reste toujours utilisé pour mémoriser le voisinage des régions et effectuer les mises à jour des adjacences après les fusions. Le critère de similarité entre deux régions

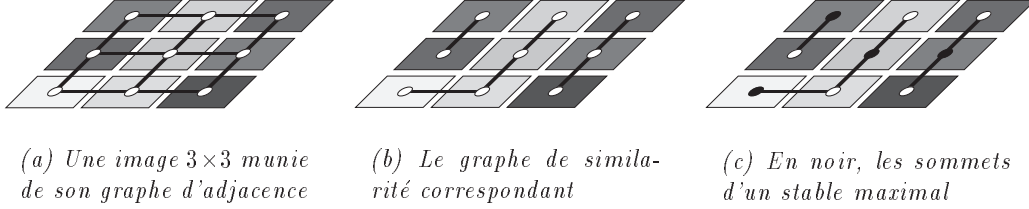


Fig. 37 - Graphe de similarité non orienté et stable maximal

voisines peut être de natures différentes ou être une combinaison de plusieurs critères simples fondés par exemple sur :

- le contraste moyen des pixels de la frontière ;
- le niveau de gris moyen des régions ;
- la distance du centre du père potentiel aux centres de ses fils potentiels [Hon84].

Nous avons utilisé un critère de similarité C_h simple. Soit $G(S, A)$ le graphe d'adjacence. et Le graphe de similarité $H(S, B)$ est défini par :

$$\forall (s_p, s_q) \in A, (s_p, s_q) \in B \text{ ssi } |g(s_p) \ominus g(s_q)| \leq T_g$$

Autrement dit, deux sommets sont similaires si leur différence de niveaux de gris est inférieure à un seuil global T_g . Ce critère est symétrique et par conséquent le graphe de similarité est non orienté. T_g doit être choisi de façon empirique par l'utilisateur. De cette valeur seuil dépendent grandement les résultats de la segmentation (§ 4.5.3). Néanmoins, une méthode propre à la pyramide irrégulière stochastique a été développée pour automatiser son calcul en fonction du contenu de l'image [Cho95].

Si T_g est fixé à sa valeur maximale (255), quel que soit le contenu de l'image, le graphe de similarité est identique au graphe d'adjacence. En revanche, dans la majorité des images traitées, fixer ce seuil dans des limites raisonnables revient à éliminer l'arête reliant 2 sommets dissimilaires voisins, évitant ainsi la fusion indésirée des 2 champs récepteurs correspondants.

Ce seuil global est une première réponse à la prise en compte du contenu de l'image. Ce critère donne de bons résultats pour des types d'images bien particuliers, mais reste limité par son manque d'adaptativité. On peut voir sur certaines images de synthèse (figure 38) que, parce que les coûts de fusions ne sont pas optimisés, certains biais peuvent être introduits par son utilisation [Mon89].

Dans l'exemple choisi, comme le dégradé est faible (deux colonnes successives c et $c + 1$ ont des niveaux de gris $g(c)$ et $g(c) + 1$) et que $T_g > |g(c) \Leftrightarrow g(c) + 1|$ (i.e. $T_g > 1$), les fusions ne se font pas dans un ordre qui maximise la similarité : au lieu de fusionner tout d'abord verticalement (jusqu'à constituer autant de régions⁶ que de colonnes de pixels), les fusions se font également horizontalement. Le phénomène s'étend sur d'autant plus de colonnes que T_g est grand.

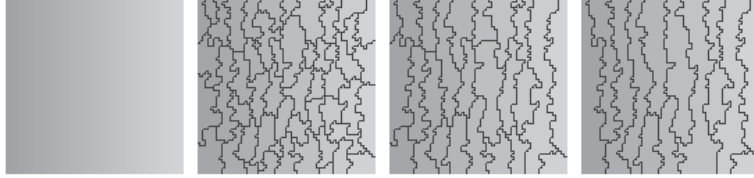


Fig. 38 - Problème posé par la similarité non orientée sur la segmentation d'un dégradé

C'est dans le but d'optimiser les coûts de fusions que Montanvert *et al* utilisent un critère de similarité défini localement. La localité des traitements génère un graphe de similarité orienté, comme l'explique le paragraphe suivant.

Graphe de similarité orienté

Un critère de similarité défini localement permet d'avoir un traitement mieux adapté aux particularités de l'image. La méthode proposée dans [Mon89] est la suivante : dans le graphe d'adjacence, chaque sommet partage son voisinage en 2 classes :

1. les voisins qui lui sont similaires ;
2. les voisins qui lui sont non similaires.

A partir du graphe d'adjacence $G(S, A)$, chaque sommet s calcule, uniquement en fonction de ses voisins $V(s)$, quel va être son propre seuil de similarité $T(s)$. Ce seuil local est la différence maximale entre son niveau de gris et le niveau de gris d'un voisin, au delà de laquelle ce voisin est considéré comme non similaire.

Vient ensuite la phase de construction du graphe de similarité. Un nouveau graphe $G'(S, \emptyset)$ est constitué, avec tous les sommets du graphe d'adjacence et aucun arc ($A' = \emptyset$). La création des arcs de A' est effectuée comme suit :

$$\forall s \in S, \forall s_i \in V(s) (s, s_i) \in A' \text{ si } |g(s_i) \Leftrightarrow g(s)| \leq T(s)$$

Autrement dit, si s_i est similaire à s_j , s_i pourra potentiellement fusionner avec s_j . Ceci, quelle que soit la similarité vue par le sommet s_j . Si s_j est aussi similaire

⁶d'écart-type nul

à s_i , la relation est symétrique et chacun des deux sommets peut potentiellement fusionner avec l'autre. En général,

$$|g(s_i) \Leftrightarrow g(s)| \leq T(s) \not\Leftrightarrow |g(s) \Leftrightarrow g(s_i)| \leq T(s_i)$$

puisque les deux seuils sont calculés à l'aide de voisinages qui ne se recouvrent que partiellement (figure 39). Notons que ce traitement peut être exécuté en parallèle sur toute l'image.

Comme le montre la figure 40, l'utilisation du seuil local permet de favoriser les meilleures fusions. la segmentation du dégradé est correcte.

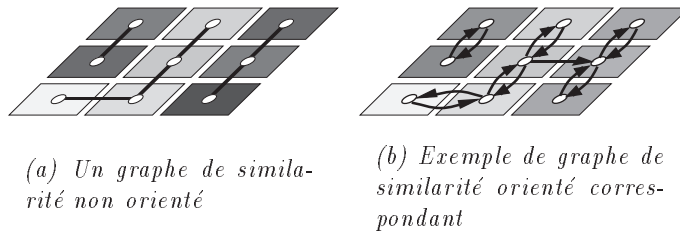


Fig. 39 - Graphe de similarité orienté

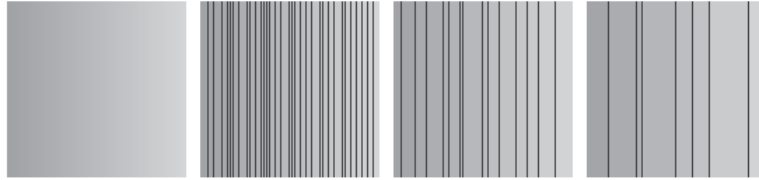


Fig. 40 - Dégradé segmenté correctement avec l'utilisation du seuil local

L'utilisation du seuil local n'entraîne pas l'abandon du seuil global T_g . T_g est en quelque sorte une limite de sécurité au delà de laquelle deux sommets adjacents sont considérés non similaires même si leur différence en niveaux de gris est inférieure au seuil local $T(s)$. Prenons le cas d'un sommet de niveau de gris clair qui ne possède que des sommets voisins de niveaux de gris foncés. Le sommet en question va partager son voisinage en deux classes : une considérée comme similaire⁷ et l'autre non, bien que tous ses voisins aient un niveau de gris très différent du sien. Il est nécessaire d'avoir un seuil absolu pour éviter les fusions erronées provoquées par un seuil local trop élevé. Cette valeur est une différence de niveaux de gris et peut donc théoriquement

⁷contenant les sommets les plus similaires d'un ensemble de sommets quoi qu'il en soit non similaires.

prendre des valeurs absolues comprises entre 0 et 255 inclus. Expérimentalement, sur les images traitées, le seuil global de similarité T_g est compris entre 4 (dans le cas extrême de l'image IRM) et 40.

La méthode de partage du voisinage d'un sommet en deux classes est la suivante : soit s un sommet et v_i ses n voisins ($0 \leq i < n$) ordonnés dans une liste, tels que

$$|g(s) \Leftrightarrow g(v_i)| \leq |g(s) \Leftrightarrow g(v_{i+1})| \quad \forall i, 0 \leq i < n \Leftrightarrow 1$$

La liste $(v_0, v_1, \dots, v_{n-2}, v_{n-1})$ peut être divisée en $n \Leftrightarrow 1$ couples de listes :

$$((v_0, \dots, v_l), (v_{l+1}, \dots, v_{n-1})) \quad \forall l, 0 \leq l < n$$

A chaque couple est associé un couple de similarités moyennes $(m_1(l), m_2(l))$ définies par :

$$m_1(l) = \sum_{i=0}^l \frac{|g(s) \Leftrightarrow g(s_i)|}{l+1}$$

$$m_2(l) = \sum_{i=l+1}^{n-1} \frac{|g(s) \Leftrightarrow g(s_i)|}{n \Leftrightarrow l \Leftrightarrow 1}$$

Le seuil local est alors défini par :

$$T(s) = \max_{l=0}^{n-1} (|m_1(l) \Leftrightarrow m_2(l)|)$$

Autrement dit, cet algorithme commence par établir une liste croissante des différences de niveaux de gris entre le sommet étudié et chacun de ses voisins. A l'aide de cette liste, on recherche la valeur pour laquelle la différence entre la moyenne des différences inférieures à ce seuil, et la moyenne des différences supérieures à ce seuil est maximale (figure 41).

Stable et noyau

L'utilisation de graphes orientés voudrait que l'on utilise une terminologie différente pour indiquer un stable maximal :

- d'une part, l'ensemble dominant est remplacé par l'**ensemble absorbant** ;
- d'autre part, le stable est remplacé par le **noyau**.

Néanmoins, nous continuons d'utiliser le terme 'stable' car dans certains cas (figure 42), il n'est pas possible, dans un graphe orienté, de trouver un noyau⁸. [Bra95] propose de relaxer la règle 1 (§ 4.2.5) pour donner :

Règle 3 *L'ensemble des survivants doit être minimal dans le sens où il doit accroître le moins possible un stable maximal pour qu'il devienne un ensemble absorbant.*

⁸La condition qui garantit la présence d'un noyau est que le graphe ne possède pas de circuit d'ordre impair.

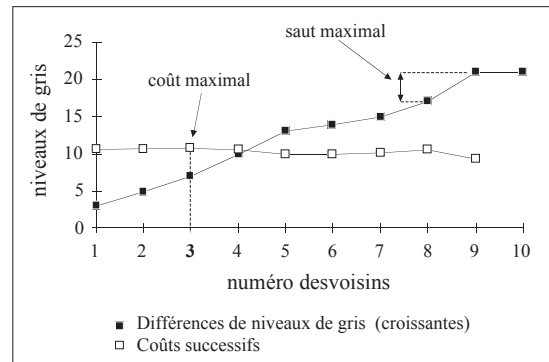


Fig. 41 - Calcul du seuil local de similarité. Le coût maximal ne correspond pas systématiquement au saut maximal. Le seuil local du sommet étudié est égal à 10,5. Seuls les sommets 1, 2, 3 et 4 sont estimés similaires au sommet étudié

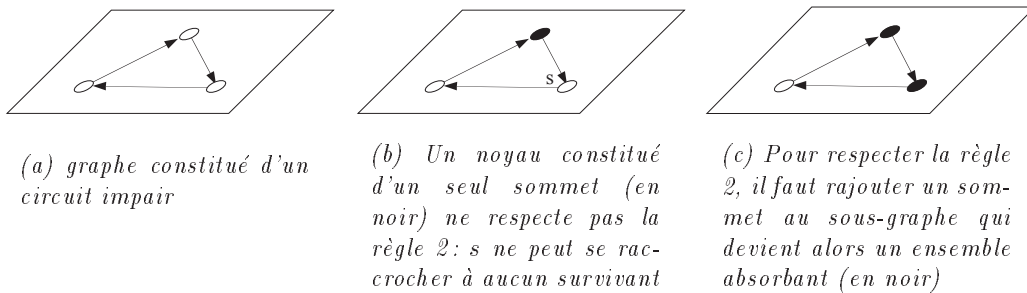


Fig. 42 - Problème posé par les graphes orientés

4.2.7 Allocation des sommets non-survivants

Nous avons décrit comment décimer les sommets d'un graphe en prenant en compte les caractéristiques de l'image (i.e. la similarité entre les régions). Nous avons défini pour chaque sommet un sous-ensemble de voisins avec lequel il peut potentiellement fusionner. Dans ce paragraphe, nous voyons comment se fait la fusion effective des régions.

Comme le graphe contracté est un stable maximal, chaque sommet non-survivant possède au moins un voisin survivant auquel se rattacher. Le but de la segmentation étant de constituer des régions homogènes, chaque sommet non-survivant va fusionner avec son voisin survivant le plus similaire (figures 43.a et b). Ces regroupements donnent lieu à la création d'un **graphe d'attachement** qui est une forêt constituée d'arbres d'attachement. Ces arbres sont soit des sommets isolés, soit des étoiles (figure 43.c). On note $\Psi(s)$ l'arbre d'attachement formé du sommet s survivant et des

non-survivants éventuels $\in V(s)$, attachés à s .

Ce traitement est effectué par la procédure **ALLOCUER NON SURVIVANTS**. Elle comporte un tirage aléatoire qui permet de choisir parmi plusieurs survivants lorsque ceux-ci sont identiquement similaires à un non-survivant voisin.

Le sommet survivant $s[k]$ (choisi par les sommets non-survivants $s_i[k]$) devient ensuite le sommet $s[k+1]$, père de $s[k]$ et de $s_i[k]$. En terme de régions, la fusion correspond au regroupement des champs récepteurs des sommets non-survivants avec le champ récepteur du survivant auquel ils se rattachent.

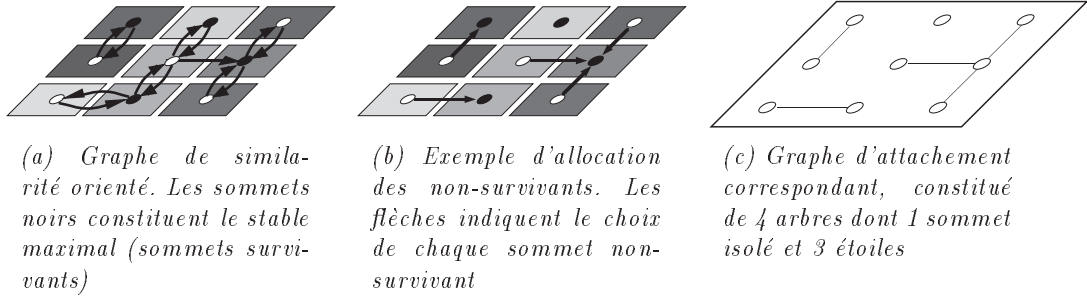


Fig. 43 - Allocation des non-survivants

4.2.8 Mise à jour des sommets survivants

Lors de la mise en place du graphe $G[k+1]$ comprenant les survivants de $G[k]$, les attributs de chaque père $p \in S[k+1]$ sont calculés à l'aide des attributs de ses fils $f_i \in S[k]$ (i.e. les attributs des sommets de son graphe d'attachement). Le traitement global correspond aux mises à jour suivantes :

- surface ;

$$S(p[k+1]) = \sum_i S(f_i[k])$$

- niveau de gris ;

$$g(p[k+1]) = \frac{\sum_i (g(f_i[k])S(f_i[k]))}{\sum_i S(f_i[k])}$$

- écart type ;

$$\sigma(p[k+1]) = \sqrt{\frac{\sum_i g(f_i[k])^2}{\sum_i S(f_i[k])}} \Leftrightarrow \left(\frac{\sum_i g(f_i[k])}{\sum_i S(f_i[k])} \right)^2$$

- champ récepteur.

$$cr(p[k+1]) = \bigcup_i cr(f_i[k])$$

4.2.9 Reconstruction du graphe d'adjacence

Nous avons vu l'extraction (à partir du niveau k) des sommets du niveau $k+1$ puis l'attachement des sommets non-survivants. Avant de réitérer le processus de construction (pour obtenir le niveau $k+2$), il reste à créer les adjacences entre les sommets du niveau $k+1$. $G[k+1]$ est créé à partir de $G[k]$ et du graphe d'attachement. La règle de création des arêtes d'adjacence est la suivante :

$$\forall s_i, s_j \in S[k+1], (s_i, s_j) \in A[k+1] \text{ssi} (\Psi(s_i), \Psi(s_j)) \in A[k]$$

Autrement dit, deux sommets survivants sont voisins au niveau $k+1$ si leurs arbres d'attachement sont voisins au niveau k (figure 44). Le voisinage entre arbres d'attachement est défini de la manière suivante : 2 arbres Ψ_i et Ψ_j sont voisins s'il existe $(s_l, s_m) \in A$ tel que $s_l \in \Psi_i$ et $s_m \in \Psi_j$. Après la création des adjacences, $G(S, A)[k+1]$ est le graphe du niveau suivant. Si $Card(G[k+1]) \neq Card(G[k])$, le traitement étudié dans ce chapitre doit alors reprendre pour définir $G[k+2]$, à commencer par la procédure **CONSTRUIRE GRAPHE DE SIMILARITE**.

4.2.10 Du graphe à l'image

La visualisation de la segmentation à chaque niveau de la pyramide passe par la reconstruction sur le support image (i.e. les pixels) de la partition à partir des informations contenues dans le graphe. Chaque sommet de l'apex définit un arbre dans la pyramide, dont les pixels sont les feuilles. Chaque pixel prend le niveau de gris de sa racine, grâce au pointeur sur son sommet racine.

4.3 Multipyramides et consensus

Le fait d'utiliser une méthode de segmentation non déterministe permet à partir d'une même image d'obtenir une grande variété de résultats qui diffèrent sensiblement. Cet aspect stochastique n'est d'ailleurs pas sans rappeler les traitements non déterministes de notre système visuel, capable de segmenter des scènes d'une façon qui diffère d'un individu à l'autre : positionnement des contours des objets, nombre d'objets, etc.

Dans un cadre général et partant de segmentations différentes d'une même image, [Mee90] tente d'extraire une réponse unique appelée image de consensus et qui approxime au mieux la segmentation statistiquement la plus probable. Cette méthode est ensuite reprise dans [Mon92a] où les auteurs construisent l'image de consensus à partir de plusieurs résultats sensiblement différents obtenus avec la pyramide

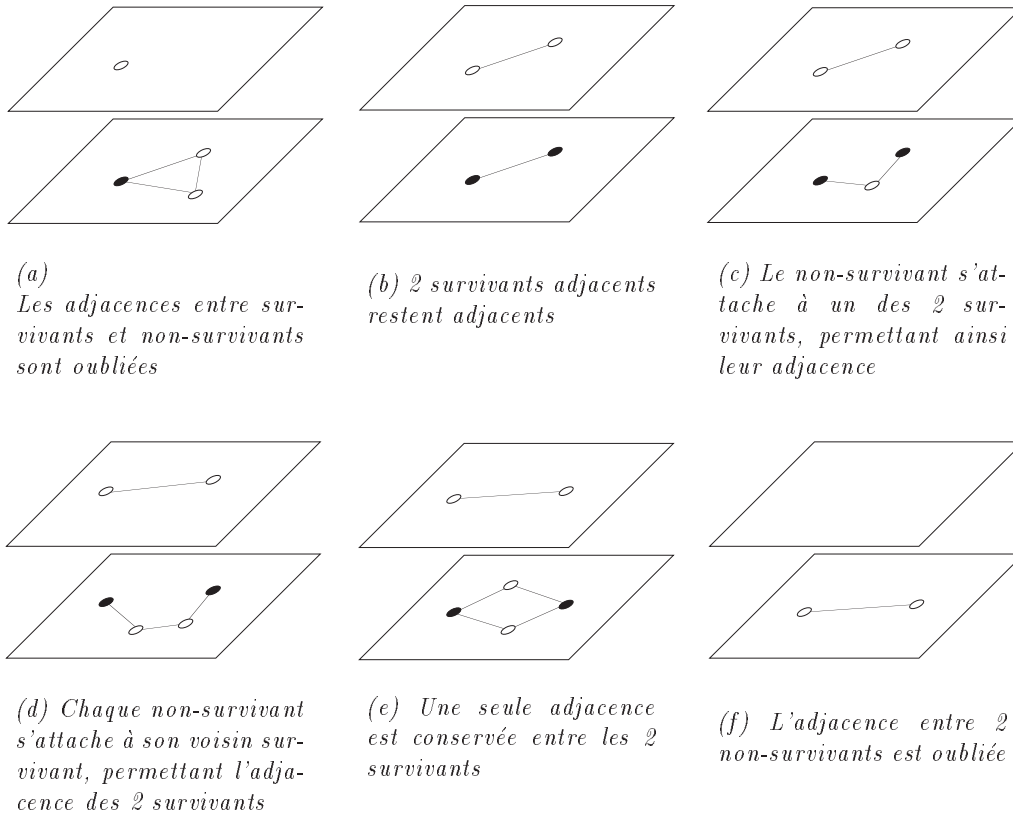


Fig. 44 - Exemple de création des adjacence du niveau supérieur. Les sommets noirs sont survivants

irrégulière stochastique. Le principe est simple :

- A partir de chaque région de l'apex, une carte de contours pondérés est établie. Sur cette carte, la valeur d'un pixel est d'autant plus élevée que ses pixels voisins n'appartiennent pas à la même région que lui : de cette façon, les pixels à l'intérieur des régions sont blancs alors que les pixels sur le bord des régions sont plus ou moins sombres (figure 45.c).
- Plusieurs cartes de contours pondérés sont calculées à l'aide de plusieurs pyramides. Puis, pour chaque coordonnée (x, y) de l'image, la valeur des pixels des cartes de contours pondérés est cumulée pour former l'image de consensus qui est en fait une accumulation d'images. Dans cette image, un pixel est d'autant plus sombre qu'il appartient à une frontière localisée au même endroit dans plusieurs images de contours pondérés. En revanche, les frontières qui apparaissent peu souvent au même endroit lors de plusieurs réalisations de la

pyramide vont donner des contours plus clairs, donc moins robustes (figure 46).

- Un seuillage simple de l'image de consensus permet d'obtenir les contours les plus probables, les plus significatifs et les plus robustes [Mon92b].



(a) Image originale

(b) Apex de la pyramide

(c) Contours pondérés

Fig. 45 - Carte de contours pondérés pour l'image de consensus



(a) Consensus de 2 images

(b) Consensus de 5 images

(c) Consensus de 10 images

(d) Consensus de 20 images

Fig. 46 - Convergence des images de consensus

4.4 Implantation

Dans cette section, nous abordons les deux techniques qui ont été utilisées pour l'implantation de la pyramide irrégulière.

4.4.1 Structure plate

Bien que la structure plate ne soit pas optimale, elle permet d'implanter la pyramide de façon assez simple tout en utilisant un minimum de place en mémoire. Le même graphe initial est mis à jour niveau après niveau. Etant donné que tous les sommets (survivants et non-survivants) restent dans la liste durant la construction de la pyramide, il convient de distinguer, avec un attribut booléen, les sommets vivants (ceux appartenant au graphe du niveau courant et devant être traités) des sommets morts (ceux n'ayant pas survécu sur l'un des niveaux précédents).

Cette structure permet une gestion particulière des champs récepteurs : la structure `sommet` possède un lien qui permet à chaque sommet (mort ou vivant) d'être relié à son sommet père (i.e. le sommet qui le représente au niveau courant). Chaque lien est remis à jour si nécessaire, niveau après niveau. Le graphe comporte ainsi des liens de voisinage et des liens de parenté (figure 47).

Lorsqu'un sommet survit sur plusieurs niveaux, ses attributs (qui au départ concernent le pixel représenté par le sommet), sont relatifs à son champ récepteur.

Lorsque l'apex de la pyramide est atteint, chacun des sommets (i.e. autant qu'il y a de pixels dans l'image) prend le niveau de gris de son sommet représentatif. La reconstruction de la composante connexe de chaque région est alors immédiate par identification des pixels portant le même label (en l'occurrence le même niveau de gris). L'utilisation de cette structure amène deux inconvénients majeurs :

1. Un seul niveau de la pyramide est connu à la fois. Aucun retour arrière n'est possible durant la construction.
2. Les temps de traitements sont élevés : le nombre de sommets du graphe reste constant de niveau en niveau. L'implantation séquentielle oblige à balayer tous les sommets pour ne traiter que les vivants.

Ces considérations ont permis de mettre en œuvre une véritable structure dynamique.

4.4.2 Structure dynamique

Dans la structure dynamique proposée, tous les niveaux construits sont disponibles simultanément. Lorsqu'un graphe de niveau k est décimé, un graphe de niveau $k + 1$ est créé, comprenant uniquement les sommets survivants du niveau k puis les arêtes d'adjacence mises à jour. (figure 34).

Ainsi, les liens de voisinage de chaque sommet (i.e. les pointeurs vers d'autres structures `sommet`) peuvent être créés dynamiquement à chaque niveau : un nombre minimal de liens (8 par exemple) est alloué à chaque sommet. Lors du rattachement des sommets non-survivants aux sommets survivants, des liens supplémentaires peuvent être alloués dynamiquement en fonction du besoin (par groupe de 8 par exemple). Par conséquent, cette méthode apporte une réponse pratique au problème théorique posé par le degré illimité des sommets. La limitation est liée

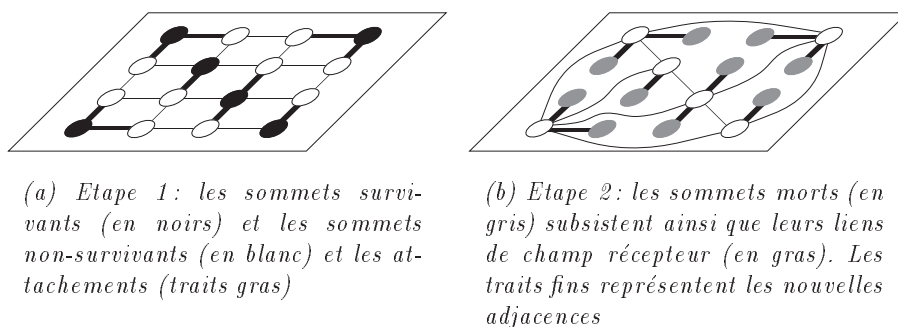


Fig. 47 - Principes de la structure pyramidale plate. Les deux figures présentent le même graphe lors de deux niveaux successifs

à la taille de la mémoire vive disponible. Les liens de voisinage des niveaux déjà construits peuvent être libérés s'ils ne sont pas réutilisés par la suite.

Le temps de traitement d'un niveau ainsi que l'espace mémoire nécessaire pour le mémoriser décroissent exponentiellement en montant dans la pyramide. Une image 256×256 est traitée en 42 sec. La construction des 3 premiers niveaux prend respectivement 16, 7 et 3 sec. Ensuite, le temps d'affichage est plus long que la construction par elle-même.

4.5 Expérimentations

Cette partie concerne l'étude expérimentale des différents mécanismes de construction de la pyramide irrégulière stochastique. Dans notre étude, les pyramides construites sur un graphe d'adjacence dépourvu d'information image⁹ sont distinguées de celles construites à partir d'une image.

4.5.1 Convergence

La construction de la pyramide s'arrête lorsque 2 niveaux successifs comportent le même nombre de régions. Respecter ce principe amène généralement deux conséquences gênantes :

1. La seconde moitié de la pyramide a un rôle presque négligeable dans le processus de réduction du nombre de régions (figure 48). En revanche, en fonction de la structure de données utilisée, les temps de traitement et la place mémoire requis peuvent être importants.

⁹Dans ce cas, l'image originale peut être assimilée à une image uniforme

2. Les fusions effectuées vers l'apex de la pyramide sont plus conséquentes que celles réalisées plus bas : le degré des voisinages faible influe alors sur la similarité. On observe souvent un phénomène de cascade néfaste : par exemple, le nombre de régions semble se stabiliser. Puis une fusion particulière relance la construction de la pyramide sur plusieurs niveaux avec généralement des fusions aberrantes.

Pour ces raisons, les résultats présentés utilisent un autre critère d'arrêt pour 'tronquer' la pyramide. L'apex de la pyramide est atteint si

$$Card(k) \Leftrightarrow Card(k+1) < \frac{Card(0)}{\alpha}$$

En pratique, nous utilisons $\alpha = 2000$. Il permet d'atteindre l'apex si 2 niveaux successifs ont moins de 2 régions de différence pour une image 64×64 , ou 32 régions de différence pour une image 256×256 . En se reportant à l'exemple étudié (figure 48) et en considérant une hauteur moyenne de 12 niveaux pour les pyramides 'tronquées', les approximations faites sont les suivantes : Soit τ_n le taux global de réduction au niveau n défini par

$$\tau_n = \frac{Card(0)}{Card(n)}$$

La comparaison de la pyramide initiale (de hauteur 19) avec la pyramide tronquée (de hauteur 12), donne $\tau_{19} = 19,5$ et $\tau_{12} = 19$. La différence du nombre de régions est de 95 soit environ 3% du nombre de régions des apex. Nous sommes ainsi convaincus que, cette approximation rend la méthode plus rapide et plus stable.

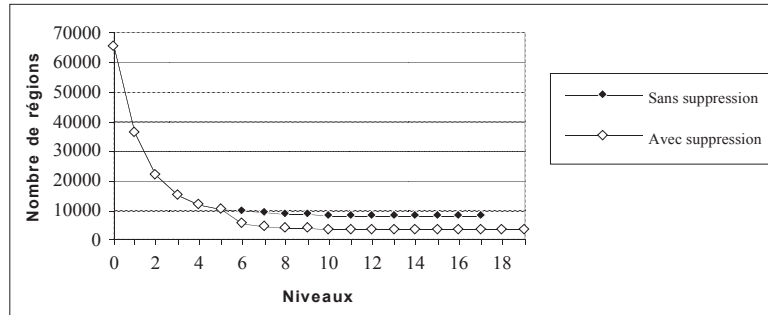


Fig. 48 - Nombre de régions en fonction des niveaux pour une segmentation de l'image 'muscle'. La pyramide supprimant les petites régions diminue nettement son nombre de régions

Pour la plupart des images, la segmentation comporte un nombre important de toutes petites régions. Ce phénomène est dû au bruit présent dans l'image et au manque de pré-traitement (lissage).

Par conséquent, pour donner aux pixels incriminés l'opportunité de fusionner naturellement avec un de leurs voisins, notre choix est de traiter ces régions à un niveau pas trop bas de la pyramide. Au cinquième niveau (la taille moyenne d'une région est alors de 5 à 20 pixels en fonction de la connexité et du seuil utilisés), le graphe de similarité est modifié de telle sorte que ces régions soient considérées similaires à tous leurs voisins. Par la suite, le processus de rattachement permettra à une région non survivante, de se rattacher à sa voisine la plus similaire.

La figure 48 montre l'effet de ce traitement sur les fusions. La différence de hauteur des 2 pyramides s'explique par le grand nombre de racines qui appauvrissent le graphe de similarité dans le cas de la pyramide sans suppression des régions d'un pixel.

Enfin, on remarque qu'en règle générale, les pyramides construites sur une 4-connexité ont une hauteur plus élevée que les pyramides construites sur une 8-connexité à raison d'un ou deux niveaux en plus dans la majorité des cas. Ceci excepté, on ne note pas de biais causé par l'utilisation d'une connexité plutôt que d'une autre.

4.5.2 Décimation

Nous nous intéressons ici au processus de décimation et à son évolution au cours de la construction des différents niveaux de la pyramide. Les résultats présentés ont été obtenus avec l'algorithme de Meer et Connely. La figure 49.a montre la différence entre les taux de décimation (ou rapport de contraction) τ obtenus sur un graphe d'adjacence G_1^{10} de 65536 sommets sur un maillage carré, avec la 4 et la 8-connexité. τ est défini par :

$$\tau[k] = \frac{\text{Card}(k \Leftrightarrow 1)}{\text{Card}(k)}$$

Ces résultats sont le moyennage de 10 itérations (pour chaque connexité) qui ont généré 10 pyramides de graphes très légèrement différentes¹¹. Notons P_4 et P_8 les 2 séries de pyramides construites sur la 4 et la 8-connexité. Il est intéressant de noter les 3 points suivants :

1. τ est plus élevé avec P_8 qu'avec P_4 pour la construction des 2 premiers niveaux (5,35 contre 2,77). Cela s'explique par le degré de voisinage qui est (en moyenne) double sur la base de la pyramide.
2. Pour P_4 et P_8 , τ a des valeurs très proches, des niveaux 3 à 7. Pour ces 5 niveaux, les topologies de P_4 et P_8 sont très ressemblantes bien que P_8 possède environ 2 fois moins de sommets que P_4 .

¹⁰sans prise en compte de l'information image

¹¹On parle également de réalisations lorsque plusieurs sorties sont dérivées d'une même entrée.

3. Dès les niveaux 8 et 9 (respectivement pour P_8 et P_4), on observe une oscillation de τ . Nous expliquons ce phénomène par le nombre trop faible de données utilisées pour calculer nos statistiques :

Toutes les réalisations de P_4 comportent au moins 9 niveaux. 7/10 des réalisations comportent 10 niveaux, 2/10 des réalisations comportent 12 niveaux.

Toutes les réalisations de P_8 comportent au moins 8 niveaux. 8/10 des réalisations comportent 9 niveaux, 3/10 des réalisations comportent 10 niveaux.

De plus, sur les derniers niveaux, les données disponibles portent sur des valeurs faibles (< 10) d'écart-type élevé.

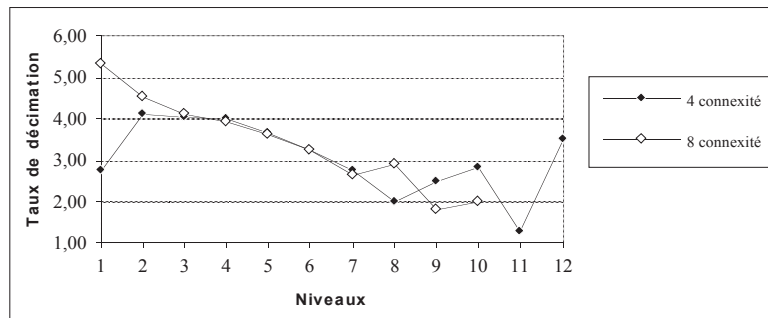


Fig. 49 - Taux de décimation sur un graphe d'adjacence comportant 65536 sommets

4.5.3 Influence du seuil global

Le résultat des segmentations dépend fortement du seuil global T_g utilisé. Plus T_g est élevé, plus le critère de similarité entre les régions est relâché et plus des régions dissimilaires ont la possibilité de fusionner entre elles. Par conséquent, le nombre de régions décroît plus vite. La figure 50 montre la décroissance du nombre de régions pour plusieurs seuils lors de la segmentation de l'image 'muscle'. On remarque les éléments suivants :

- La hauteur de la pyramide (tronquée) est peu sensible à la valeur du seuil.
- Le nombre de régions augmente exponentiellement lorsque le seuil diminue linéairement.
- Le saut entre les niveaux 5 et 6 correspondant à la prise en compte des régions mono-pixel, est d'autant plus faible que le seuil est élevé. En effet, pour T_g élevé, les régions de ce type sont moins nombreuses car elles ont une probabilité plus grande de trouver une région voisine similaire vers le bas de la pyramide.

La dépendance au seuil global est illustrée par la figure 51 qui montre des segmentations de l'image 'muscle' pour 3 T_g différents.

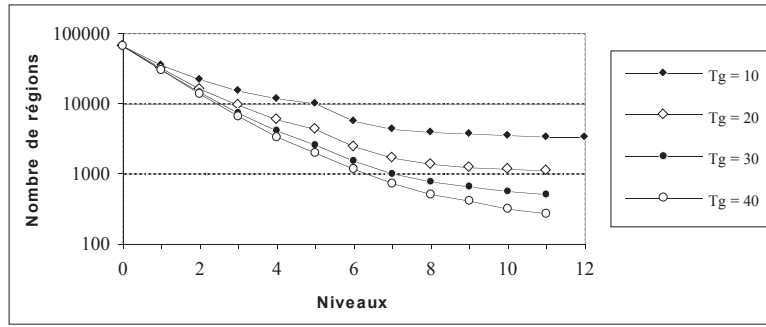
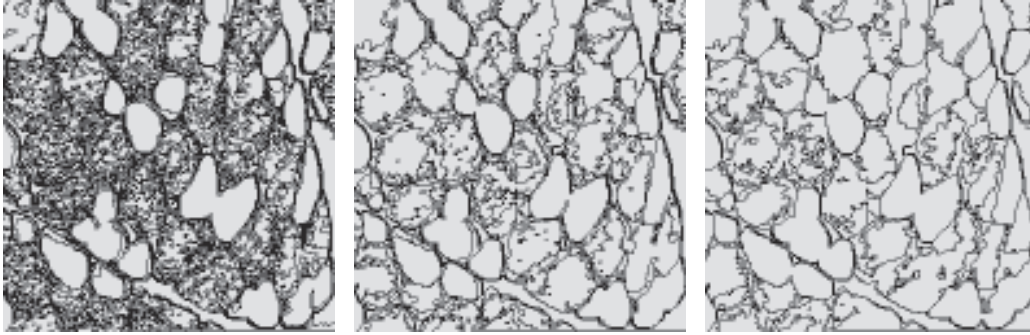


Fig. 50 - Nombre de régions par niveau en fonction du seuil global, pour la segmentation de l'image 'muscle'



(a) $T_g = 10$, 3456 régions (b) $T_g = 20$, 1126 régions (c) $T_g = 30$, 513 régions

Fig. 51 - Segmentations pour différents seuils de l'image. 'muscle', en 4-connexité

4.5.4 Degré des sommets

Le degré théorique d'un sommet de la pyramide irrégulière est non borné. Il dépend très fortement du contenu de l'image comme le montre à titre d'exemple la figure 52. Le degré moyen reste faible dans tous les cas, alors que le degré maximal peut varier de façon chaotique. Sur l'exemple donné en figure 52.a, il est obtenu pour une image prise en 4-connexité avec une valeur de 80. En revanche, la figure 52.b montre que le voisinage reste borné (inférieur à 10) dès lors que la pyramide est construite sur un graphe d'adjacence qui permet de respecter une distribution équitablement répartie des sommets survivants.

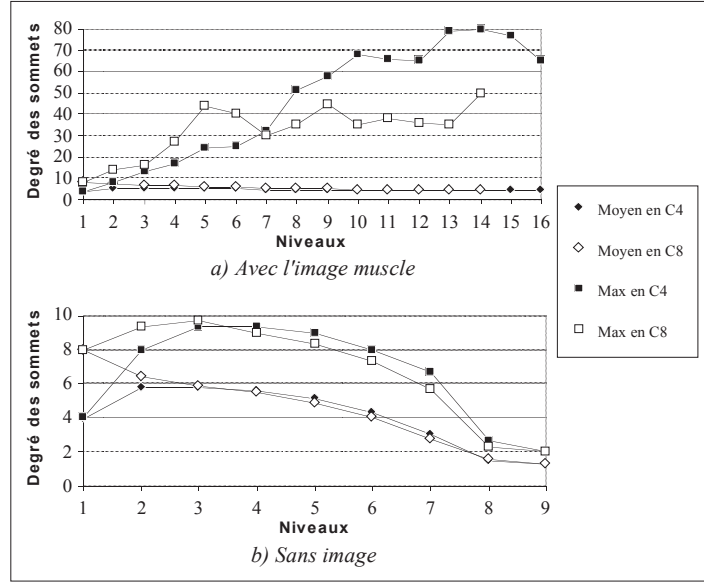


Fig. 52 - Evolution du degré des sommets pour la pyramide de l'image 'muscle' (a) et pour la pyramide d'une image uniforme (b). Les valeurs moyennes des degrés ainsi que les valeurs maximales sont données pour un graphe initial en 4 ou 8-connerité

4.5.5 Influence des tirages aléatoires

Nous remarquons (figure 53) l'effet des tirages aléatoires sur la non-répétabilité des résultats. Nous nous intéressons particulièrement à la variabilité des résultats pour un même niveau et à l'évolution de cette variabilité au cours des différents niveaux.

Etant donné la grande différence entre le nombre de régions des différents niveaux, on peut relativiser leur écart-type en calculant les coefficients de variations associés. Ceux-ci donnent alors l'écart en pourcentage par rapport à la moyenne :

$$CV[k] = \frac{\sigma(Card(k))}{M(Card(k))}$$

Dans le cas de pyramides de graphes d'adjacence (i.e. sans contenu image), le CV reste assez faible sur les 3 premiers niveaux (inférieur à 4%). Par la suite, il augmente assez fortement pour atteindre 57% pour $P4$ au niveau 9 et 36% pour $P8$ au niveau 8¹² (figure 54.a). Cette disparité s'explique par le faible nombre de régions sur ces niveaux où une différence de quelques régions entraîne une variation élevée en pourcentage.

¹²toutes les réalisations ont au moins cette hauteur

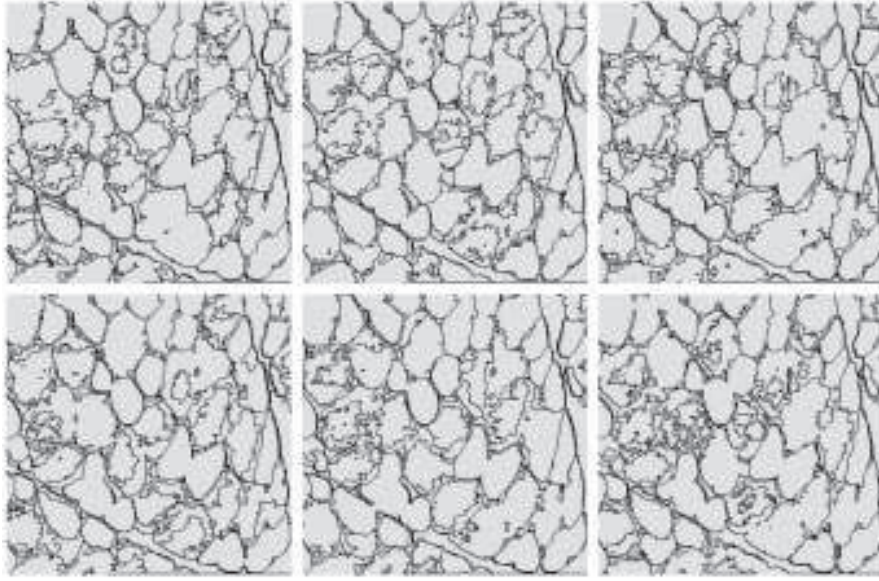


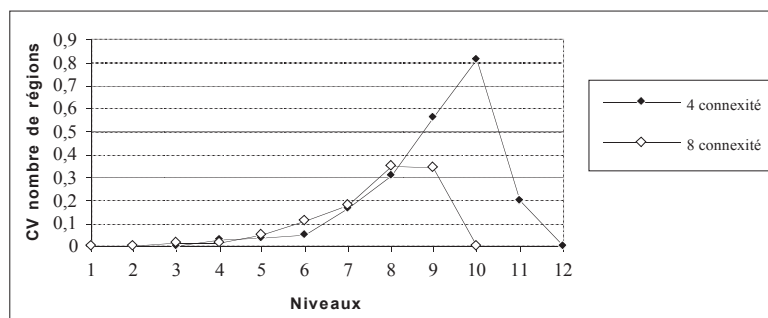
Fig. 53 - Influence du tirage aléatoire sur le résultat de la segmentation. Les apex comportent respectivement 487, 476, 505, 475, 492 et 479 régions. $T_g = 30$, 4-connexité

Dans le cas de pyramides d'images (construites sur l'image 'muscle'), les CV ont des valeurs beaucoup plus faibles (inférieures à 1% pour les 3 premiers niveaux (figure 54.b)). La valeur culmine à 3% à l'apex. La chute du CV au niveau 6 est due à la prise en compte des petites régions. Un pourcentage élevé de fusions est dû à leur traitement (comme on peut aussi le voir figure 48). Le phénomène de fusion de ces régions constituées d'un pixel est assez robuste entre plusieurs réalisations. Cette robustesse réduit ponctuellement la variabilité des réalisations.

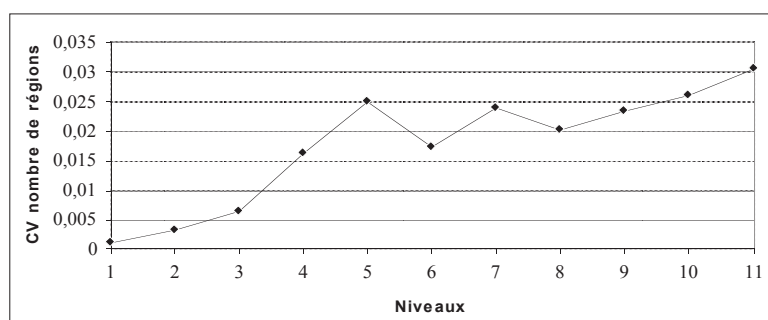
4.5.6 Résultats expérimentaux

Des résultats comparatifs entre méthodes sont proposés au chapitre 7. Afin de discuter qualitativement sur les résultats obtenus avec la méthode étudiée, nous avons choisi l'image 'muscle' figure 55. Le traitement permettant d'obtenir les images de cette figure a été initialisé avec un seuil $T_g = 35$, en 8-connexité. On remarque que les fusions sont de très bonne qualité jusqu'au niveau 4, et que la croissance se fait de façon homogène, avec tout de même des vitesses différentes dans le fond très clair, les fibres sombres et les fibres claires.

Au niveau 4, malgré un rapport de contraction élevé (de l'ordre de 33 depuis la base de la pyramide), aucune fusion erronée n'est présente. Par la suite, ce type de fusions apparaît. Les cercles blancs indiquent des frontières sémantiquement correctes qui disparaissent au niveau suivant. Dans ce cas, les fibres sombres sont défavorisées



(a) Pyramides de graphes d'adjacence. Les CV sont élevés



(b) Pyramides construites sur l'image muscle

Fig. 54 - Robustesse de la décimation. Les graphiques indiquent le pourcentage de variation du nombre de régions pour chaque niveau, lors de la construction de 6 pyramides

car les discontinuités qui les séparent sont beaucoup plus faibles que T_g . L'utilisation du seuil local permet dans un premier temps une croissance de ces régions à l'intérieur de chacune de ces fibres. Ensuite¹³, la fusion est inévitable entre fibres voisines.

La croissance dans les fibres claires est beaucoup plus anarchique. Ceci est dû à leur forte texture. Le passage du niveau 14 au niveau 15 voit la perte du peu d'information qui subsistait sur les fibres claires.

On observe un autre phénomène : à l'intérieur de plusieurs fibres claires, la croissance reste bloquée depuis le niveau 9 jusqu'à l'apex, sans entité correspondante visible.

Le résultat final comporte un certain nombre de régions bien segmentées (la moitié

¹³en dernier recours, en quelque sorte

supérieure droite de l'image). Un post-traitement spécifique permettrait sans doute de récupérer l'intégrité des cellules sombres. En revanche, dans le cas des cellules claires, l'information est très dégradée depuis de nombreux niveaux, et la mauvaise segmentation est irréversible.

4.6 Discussion et conclusion

Au cours de la construction de la pyramide, un double phénomène préjudiciable à la segmentation est observé :

- d'une part, des discontinuités significatives sur l'image originale ont tendance à disparaître ;
- d'autre part, des frontières apparaissent dans des régions assez homogènes. Ce phénomène incontournable est également constaté sur l'image de synthèse de la figure 56 qui présente une suite de dégradés identiques.

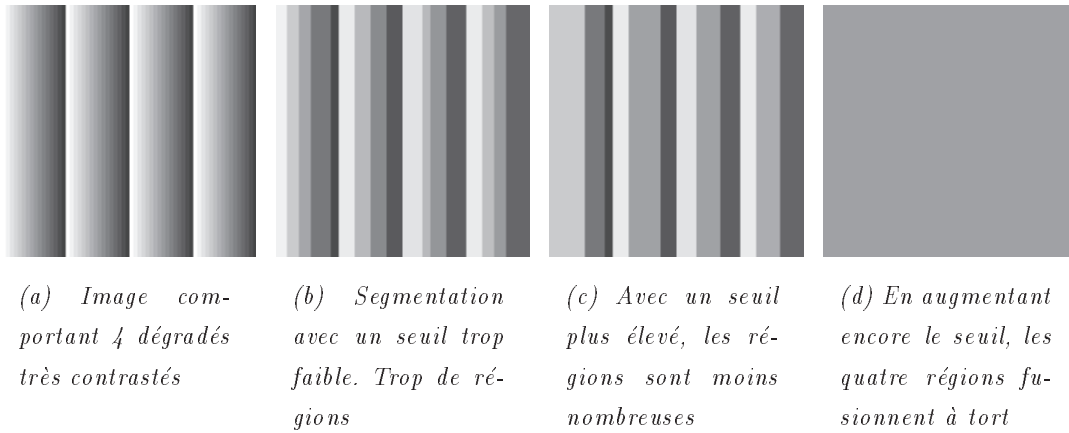


Fig. 56 - Problématique de la pyramide irrégulière sur une image de synthèse

Une segmentation correcte devrait fournir quatre régions. En effet, l'image représente quatre dégradés parfaitement continus, séparés par de fortes discontinuités. La construction de la pyramide se déroule de la façon suivante : les fusions s'opèrent tout d'abord verticalement par colonnes de pixels pour satisfaire le critère de similarité C_h évalué sur le seuil adaptatif. Lorsque chaque colonne de pixel est construite, les fusions se poursuivent horizontalement entre colonnes voisines. Le niveau de gris de chaque région est recalculé : il prend la valeur moyenne de l'ensemble des pixels qui la composent. On observe alors une divergence des niveaux de gris entre régions voisines. Lorsque la valeur de la divergence est supérieure à T_g , toute fusion est

rendue impossible. Si on augmente la valeur de T_g , la largeur des bandes verticales augmente alors que le nombre de régions diminue. Dès que T_g est assez élevé pour que toutes les régions formant un dégradé fusionnent, les quatre régions correspondant aux quatre dégradés de l'image fusionnent également car elles possèdent toutes le même niveau de gris moyen.

Dans le chapitre suivant, nous proposerons une méthode qui permet de résoudre ce problème d'école et d'améliorer la segmentation d'images naturelles.

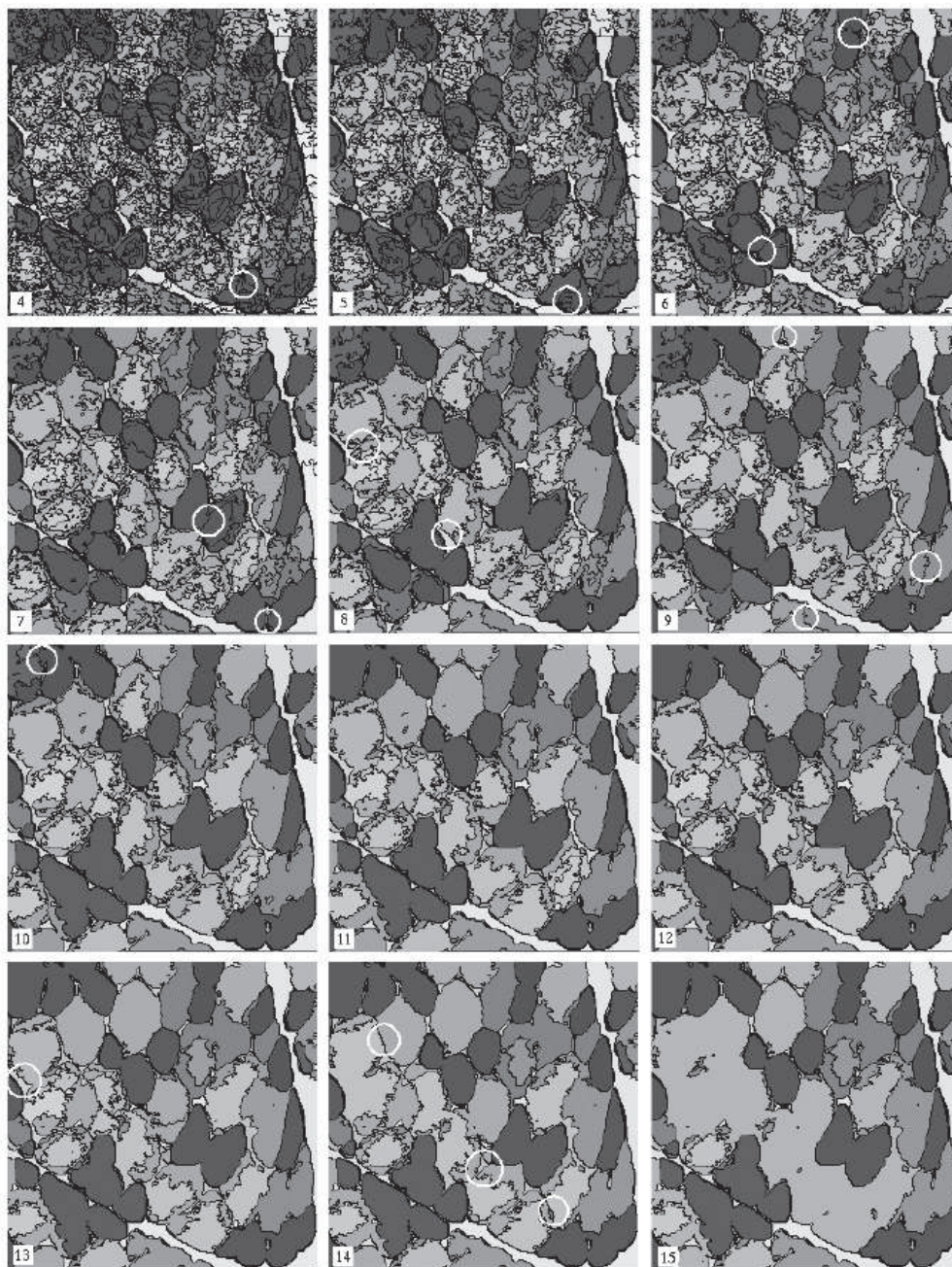


Fig. 55 - Pyramide irrégulière de l'image 'muscle'. Seuls les niveaux de 4 à 15 sont représentés. Les contours des régions sont marqués en noir. Les cercles blancs indiquent la limite de 2 régions qui vont fusionner à tort dans le niveau supérieur

Chapitre 5

Coopération régions-contours

En ce qui concerne les résultats de segmentation, nous avons vu les problèmes posés par les fusions erronées. Nous avons noté également que, dans plusieurs images, des fusions pertinentes étaient progressivement rendues impossibles.

Pour pallier ce problème, nous introduisons des notions autres que celles qui mettent en œuvre un critère de similarité entre des régions adjacentes. Notre idée consiste à déterminer les contours significatifs à la base de la pyramide, à transmettre et à faire évoluer cette information à tous les niveaux de la construction. Cette information complémentaire permettra d'interdire des fusions erronées et de forcer des fusions correctes.

5.1 Principes de coopération

Le graphe de similarité décrit au chapitre précédent (§ 4.2.6) est cohérent. Aussi, dans le travail présenté, son utilisation est conservée. Un mécanisme complémentaire est développé pour mettre à jour ce graphe ponctuellement, afin de prendre d'avantage en compte les informations présentes dans l'image. La méthode doit donner les possibilités suivantes :

1. Fusionner 2 régions non similaires. C'est le principe de la **fusion imposée**.
2. Empêcher la fusion de 2 régions similaires. C'est le principe de la **fusion empêchée**.

Ces principes doivent intervenir à bon escient, lorsque le graphe de similarité (construit à l'aide du critère de similarité C_h) véhicule une information qui est obsolète, comme dans l'exemple du dégradé en § 4.6.

L'information utilisée pour mettre à jour le graphe de similarité va être stockée sur le graphe d'adjacence. En effet, chacune des arête peut être pondérée pour représenter un certain nombre d'attributs propres à décrire qualitativement et quantitativement cette adjacence.

5.2 Coopération réduite

Dans une première approche expérimentale, nous avons rajouté un attribut (appelé **qualité de voisinage** Q) quantifiant les discontinuités (éléments de contours) présentes sur l'image originale. Q pondère chaque arête sous la forme d'une information binaire. C'est une fonction de la différence du niveau de gris de deux sommets adjacents s_i et s_j . Soit C (coût), cette différence. On a :

$$C(s_i, s_j) = |g(s_i) \leftrightarrow g(s_j)|$$

Q est initialisée de la manière suivante : Soit T_g le seuil global de similarité. Entre deux sommets s_i et s_j , $Q(s_i, s_j) = Q(s_j, s_i)$ et peut prendre deux valeurs :

$$Q(s_i, s_j) = \begin{cases} 1 & \text{si } C(s_i, s_j) < T_g \\ 0 & \text{sinon} \end{cases}$$

L'utilisation de Q est faite à travers l'évaluation d'un nouveau critère C_d chargé de déterminer les discontinuités entre régions. Q est utilisée de deux façons différentes :

1. d'une part pour empêcher des fusions valides au sens de C_h et invalides au sens de C_d ;
2. d'autre part pour provoquer des fusions invalides au sens de C_h et valides au sens de C_d .

La gestion du nouvel attribut $Q(s_i, s_j)$ et du nouveau critère C_d a été implantée. Plusieurs problèmes inhérents à l'aspect binaire de cette information sont apparus :

- A l'initialisation tout d'abord : à quel niveau k_{init} de la pyramide faut-il initialiser les $Q(s_i, s_j)$? Si k_{init} est trop bas ($k_{init} = 0$ par exemple), toutes les fortes discontinuités de l'image dues au phénomène de bruit sont prises en compte comme des éléments de contour. Si k_{init} est trop élevé, certaines discontinuités significatives entre sommets auront déjà disparu ;
- A l'utilisation ensuite : comment doivent être choisis le (ou les) niveau(x) k_{util} pour le(s)quel(s) les fusions imposées et empêchées doivent intervenir en utilisant les pondérations ?
- Enfin, il existe un problème à la mise à jour : quelle doit être la logique établissant la mise à jour des $Q(s_i, s_j)$ du niveau $k + 1$ en fonction des $Q(s_i, s_j)$ du niveau k ?

Nous n'avons trouvé aucune réponse réellement satisfaisante à une des trois questions énoncées ci-dessus. D'une part, k_{init} et k_{util} paraissent très dépendants de l'image traitée et ne donnent pas de résultats satisfaisants. D'autre part, la mise à jour doit s'effectuer sur des données binaires non quantifiables, induisant des prises de décisions trop expéditives.

De cette première approche de coopération régions-contours, nous conservons par la suite les notions de qualité de voisinage et de critère de discontinuité. Dans le paragraphe suivant, nous développons une coopération plus rationnelle et formalisée.

5.3 Coopération étendue

Dans la méthode de coopération présentée [Ber96], nous distinguons 4 phases :

1. le calcul d'une carte d'éléments de contour I_c de l'image originale ;
2. la pondération du graphe d'adjacence de la base de la pyramide par les informations données par I_c ;
3. la mise à jour d'un niveau à l'autre des pondérations du graphe d'adjacence ;
4. l'utilisation des pondérations qui vont
 - tout d'abord empêcher certaines fusions ;
 - ensuite provoquer certaines fusions.

Après avoir commenté ces différentes phases, nous choisirons un détecteur de contours approprié.

5.3.1 Pondération du graphe d'adjacence

La carte des contours utilisée doit être binaire. Comme il s'agit de pondérer les arêtes du graphe d'adjacence induit par la 4 ou 8-connexité des pixels, ces contours doivent être du type 'processus de ligne' (§ 3.3.2) pour décrire une discontinuité prenant en compte 2 pixels voisins. Par la suite, cette discontinuité est appelée **élément de contour** par opposition à un **élément de région** constitué de 2 pixels similaires.

La carte des éléments de contour est mise en correspondance (pixel à pixel) avec le graphe d'adjacence de l'image initiale (figure 57.a et b). Chaque arête est pondérée avec un couple de valeurs (r, c) , r et c représentant respectivement la quantité d'éléments de région et d'éléments de contour de la frontière commune à deux régions. Toutes les arêtes du graphe correspondant à un élément de la carte des éléments de contours sont initialisées avec le couple $(r, c) = (0, 1)$. Inversement, toutes les arêtes du graphe qui ne correspondent pas à un élément de contour sont initialisées avec le couple $(r, c) = (1, 0)$ (figure 57.c).

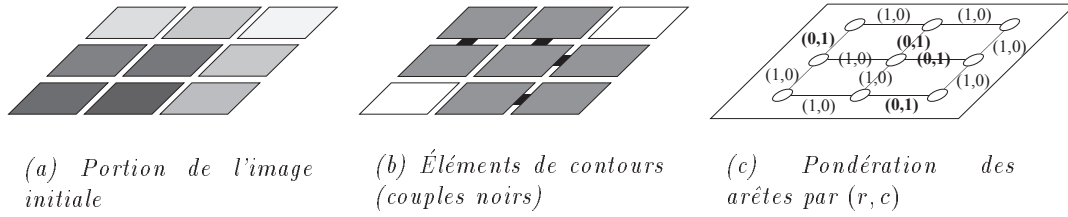


Fig. 57 - Initialisation des pondérations du graphe d'adjacence

La pondération du graphe permet désormais de connaître la quantité de discontinuité entre deux régions par rapport à la longueur totale de la frontière commune à ces deux régions (i.e. $r + c$). Après la phase d'initialisation la pyramide irrégulière est construite et utilisée comme auparavant. Cependant, la mise à jour des pondérations (r, c) et leur utilisation pour affiner le graphe de similarité se feront à chaque niveau.

5.3.2 Mise à jour du graphe de similarité

A la construction de chaque niveau, le graphe de similarité $G'(k)$ est dans un premier temps mis à jour comme précédemment, c'est-à-dire à l'aide du seuil de similarité global et du critère de similarité local.

Ensuite, un critère de discontinuité C_d détermine si la pondération d'une arête par (r, c) conduit à interdire ou à favoriser une fusion : le critère C_d permet d'enlever ou de rajouter des arcs sur le graphe de similarité établi à l'aide du critère C_h . Notons que pour deux sommets adjacents v_i et v_j , $(r, c)_{v_i} = (r, c)_{v_j}$: le critère C_d est

symétrique contrairement au critère C_h . C_d est défini de la manière suivante :

- A chaque niveau de la pyramide, le (ou les) arcs entre deux régions sont enlevés de $G'(k)$ si le nombre d'éléments de contours est trop élevé par rapport au nombre d'éléments de régions. L'évaluation choisie consiste à ôter un arc si $\alpha c > r$. α est un coefficient pondérateur qui permet de prendre en compte d'une façon plus ou moins importante les informations de discontinuité par rapport aux informations d'homogénéité. Si $\alpha > 1$, les contours utilisés jouent un rôle important dans le processus de coopération : un nombre réduit d'éléments de contour sur une frontière homogène entre deux régions interdit une fusion possible au sens du critère C_h .
- Lorsqu'il n'y a plus d'arc dans le graphe de similarité $G'(k)$ (i.e. l'apex de la pyramide est atteint), l'information (r, c) permet d'en créer. Cela se produit lorsque deux régions de niveau de gris non similaires au sens de C_h ¹ ne comportent pas de frontière significative sur l'image originale : deux arcs sont rajoutés entre deux sommets si le nombre d'éléments de région est assez élevé par rapport au nombre d'éléments de contour. L'évaluation choisie consiste à rajouter des arêtes si $\alpha c \leq r$.

5.3.3 Mise à jour des pondérations

La mise à jour effectuée entre les niveaux k et $k+1$ est simple (figure 58) : une arête du niveau $k+1$ représente une ou plusieurs arêtes i du niveau k . Par conséquent, la mise à jour des pondérations d'une arête s'écrit :

$$(r, c)_{k+1} = \left(\sum_i r_i, \sum_i c_i \right)_k$$

L'information de frontière entre sommets du niveau k qui fusionnent entre eux n'est plus nécessaire au niveau $k+1$ et n'est donc pas transmise dans la pyramide (figure 58).

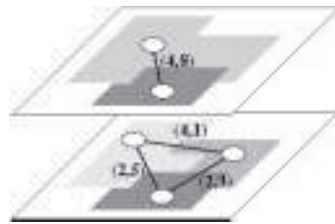


Fig. 58 - Exemple de mise à jour inter-niveaux des pondérations

¹C'est le cas par exemple lors de la segmentation d'une zone dégradée

5.4 Utilisation de détecteurs existants

Dans notre approche, les contours nécessaires pour pondérer les arêtes du graphe d'adjacence peuvent être obtenus à partir de détecteurs fonctionnant sur la recherche des maximums locaux de la norme du gradient (§ 3.3.1).

5.4.1 Recherche des sous-maximums locaux

Les détecteurs délivrent des cartes de contours d'épaisseur 1 pixel, il faut donc chercher, pour chaque pixel de contour, son voisin le plus apte pour former un élément de contour de 2 pixels.

Les 2 éléments de contours ont une probabilité maximale d'appartenir à 2 régions voisines si le couple s'obtient en recherchant, pour chaque pixel de la carte des contours, son voisin dans la direction des gradients dont la norme du gradient est maximale (sous-maximum local). La recherche d'un sous-maximum s'effectue dans les directions compatibles avec la connexité choisie pour initialiser la pyramide : verticale, horizontale et diagonales pour la 8-connexité ; verticale et horizontale pour la 4-connexité.

Notons que cette recherche est peu coûteuse puisqu'elle utilise la carte des maximums locaux seuillés, la carte des normes des gradients et la carte de la direction des gradients qui sont des résultats déjà fournis par la majorité des détecteurs de contours.

La figure 59.a montre avec deux couleurs différentes les maximums extraits avec le détecteur de Deriche et les sous-maximums associés. Des résultats comparatifs entre méthodes sont proposés au chapitre 7. Une approche intéressante consisterait à utiliser les maximums de l'image de consensus obtenue à partir de plusieurs pyramides stochastiques. Cette image donne des informations intéressantes sur la localisation et la force des contours dans l'image.

5.4.2 Recherche des éléments de contour

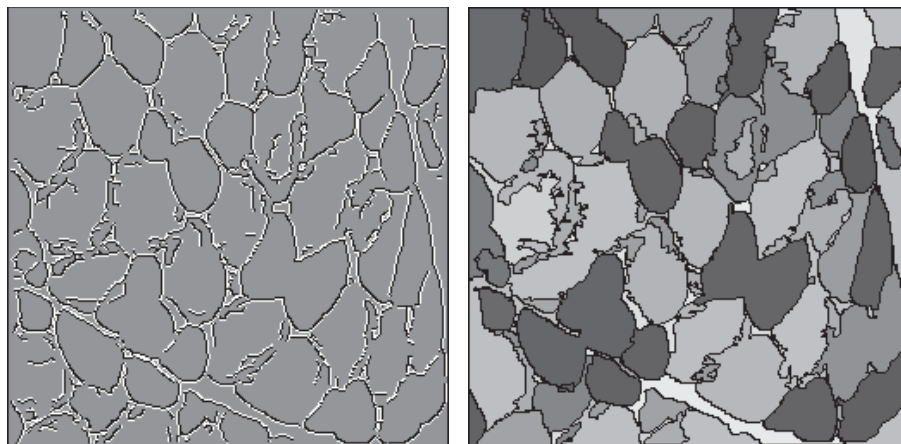
L'association de chaque maximum local à un sous-maximum local ne suffit pas à pondérer complètement le graphe d'adjacence même lorsque le contour est continu. Une pondération non complète peut induire des 'fuites' entre régions.

Soit l , la longueur de la frontière entre 2 régions (c'est à dire le nombre d'arêtes qui leurs sont communes) et n le nombre d'éléments de contour entre ces 2 régions.

Lorsque le contour est vertical ou horizontal, $l = n$, il y a autant d'éléments de contour que d'arêtes d'adjacence entre les 2 régions potentielles et l'étanchéité est garantie. (figure 60.a).

Lorsque le contour est diagonal (ou change de direction), $n < l$. Il faut pondérer toutes les arêtes entre maximums et sous-maximums adjacents comme des arêtes correspondant à des éléments de contour (figure 60.b).

Néanmoins, cette condition ne suffit pas toujours à obtenir l'égalité $l = n$, que ce soit en 4 ou en 8-connexité. Une solution consiste à étudier de façon précise le



(a) Carte des éléments de contours
(chaque élément est composé d'un
pixel noir et d'un pixel blanc)

(b) Résultat de la segmentation

Fig. 59 - Coopération région-contour avec l'opérateur de Deriche

voisinage de chaque maximum afin de choisir le nombre de sous-maximums voulus jusqu'à étanchéité complète.

5.5 Détection de contours par lancer de segments

Les raisons pour lesquelles nous avons développé un détecteur de contours spécifique à la coopération sont les suivantes :

- Les détecteurs fournissent des contours qui sont assez continus lorsque les discontinuités sont fortes, et absents lorsqu'elles sont faibles. La croissance de régions en parallèle de la pyramide gère assez bien les fortes discontinuités. En revanche, elle nécessite des informations de contours lorsque les discontinuités sont faibles.
- Les contours utilisés pour la coopération doivent être nombreux même s'ils ne sont pas continus. La croissance de régions doit pouvoir s'appuyer localement sur des éléments de contours plus qualitatifs que quantitatifs.
- Les détecteurs de contours les plus performants nécessitent un paramétrage complexe, adapté à chaque type d'image. Nous désirons une méthode plus adaptative.

Le détecteur de contours spécifique que nous avons mis en œuvre permet de conserver toutes les propriétés inhérentes à la structure pyramidale irrégulière et à

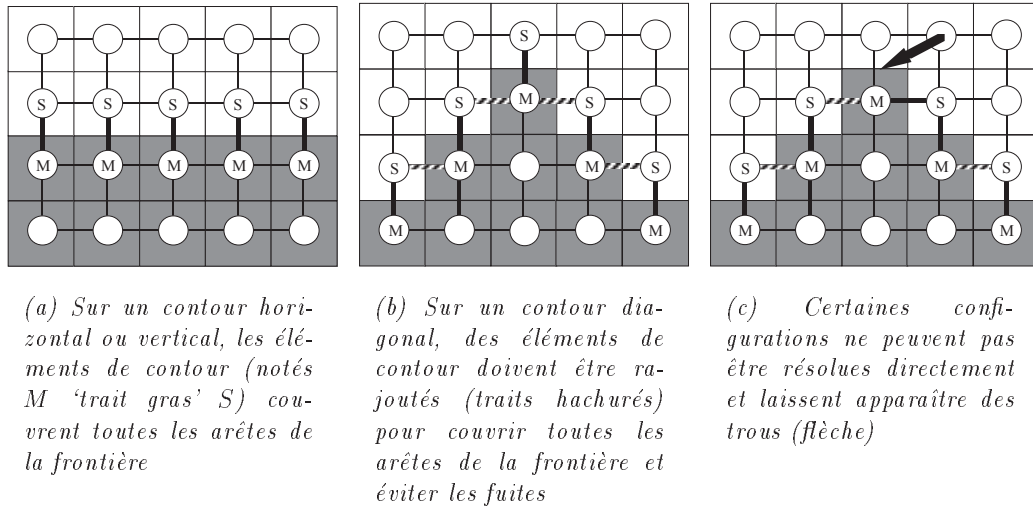


Fig. 60 - Problème d'extraction des éléments de contour sur une image de deux régions contrastées en 4-connexité

sa modélisation [Mon92b].

5.5.1 Opérateur 1D vs opérateurs 2D

Ce détecteur de contours consiste à trouver dans l'image, des couples de pixels adjacents qui constituent localement un contraste significatif. Chacun de ces couples forme ainsi un élément de contour élémentaire qui a une probabilité élevée d'appartenir à une chaîne de contour. Contrairement aux détecteurs de contours traditionnels, celui-ci délivre un ensemble de couples de pixels plutôt que des chaînes de pixels.

En considérant un élément de contour sous un angle relatif à l'approche 'régions', nous émettons l'hypothèse que les deux pixels de cet élément appartiennent à deux régions différentes (figure 61). Grâce à ce lien de structure entre régions et élément de contour, une coopération possible entre ces entités peut être établie.

Les détecteurs de contours classiques utilisent généralement comme support de base à leur recherche, une fenêtre rectangulaire d'une taille donnée qui est déplacée dans l'image. Une telle décomposition de l'image permet des traitements locaux qui dégagent des caractéristiques en utilisant une information spatiale de dimension 2. Cependant, l'utilisation de sous-ensembles données par les fenêtres soulèvent deux problèmes majeurs :

1. La taille pré-définie des fenêtres (par exemple les masques de convolution Laplaciens 5×5) qui correspondent à un seuil implicite.

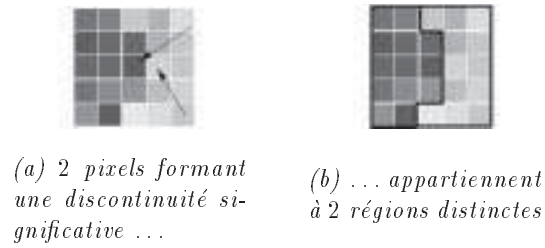


Fig. 61 - Relation entre régions et contours

2. La complexité d'utiliser des outils et des critères appropriés pour étudier des environnements 2D [Pav92].

Notre algorithme détecte des caractéristiques sur un support monodimensionnel : le segment de droite. Il y a quelques années, une approche de type monodimensionnel a déjà été proposée pour effectuer un seuillage adaptatif sur l'image [Qui91] : les auteurs utilisent des caméras ligne qui fournissent un signal monodimensionnel de 512 octets. Pour chaque ligne, leur algorithme extrait toutes les déclivités (variations de niveaux de gris entre deux extremums locaux). Cet ensemble de déclivités leur permet de construire un histogramme qui donne leur nombre en fonction de leur amplitude. Cet histogramme est en général bimodal. Leurs expérimentations fournissent l'heuristique suivante : la vallée située entre les deux modes correspond au seuil à adopter pour la ligne traitée. Notre approche utilise un principe proche, mais stochastique.

5.5.2 Principe du détecteur

Une image numérique I est une matrice de pixels sur une grille où peut être définie un ensemble fini S de tous les segments discrets possibles (figure 62). Pour une image de taille $N \times N$, le nombre de segments de taille l ($1 < l \leq N$) est

$$\text{Card}(S) = \frac{N^2 \times (N^2 \Leftrightarrow 1)}{2}$$

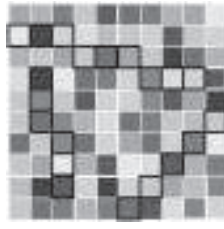
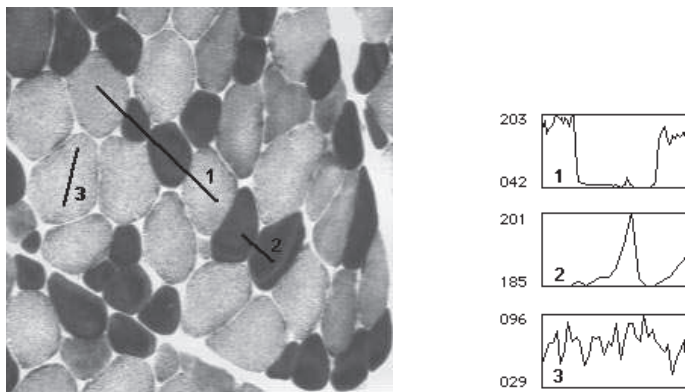


Fig. 62 - Exemples de 3 segments

Chaque segment s de longueur l appartenant à S est un ensemble connexe de pixels ordonnés, numérotés de 0 à $l \Leftrightarrow 1$: $s = s_0, \dots, s_{l-1}$. Le sens de parcours du segment, et donc la façon dont les pixels sont numérotés est arbitraire. Une correspondance est définie entre S et F , F étant l'ensemble des fonctions discrètes $f(x), (x = 0, \dots, l \Leftrightarrow 1)$ de dimension 1 appelées profils (figure 63). Ce sont les fonctions de niveaux de gris des pixels qui appartiennent au segment correspondant de longueur l .



(a) Segments traversant des discontinuités fortes (1), faible (2) et du bruit (3)

(b) Les profils correspondants montrent des discontinuités significatives pour (1) et (2)

Fig. 63 - Segments et profils divers

Le long de chaque segment $s \in S$, et donc pour chaque fonction $f \in F$, nous voulons rechercher s'il existe une caractéristique significative. Le nombre de segments qui traversent un pixel est en $O(N^2)$. La longueur de ces segments varie de 2 à N . De cette façon, la contribution et la valeur de chaque pixel (à une zone homogène ou hétérogène) sont étudiées à travers de nombreuses résolutions et distributions spatiales. Traiter $Card(S)$ de manière exhaustive n'est pas envisageable ($Card(S) =$

Algorithme 8 (*Détecteur stochastique de contours*)

```

début
  Carte_des_contours <- ∅;
  Répéter un grand nombre de fois
    début
      Choisir aléatoirement 1 point A dans l'image ;
      Choisir aléatoirement un angle a dans [0, 2 PI] ;
      Choisir aléatoirement une longueur l dans [3, M] ;
      Calculer B, le point situé à distance l de A, sur une direction a ;
      Construire le segment s = [A, B];
      Déduire f de s;
      Chercher (f(i) et f(i+1)) qui vérifient C ;
      si ((f(i); f(i+1)) existe) alors
        Ajouter l'élément de contour (s_i, s_i+1) à Carte_des_contours;
    fin;
fin.

```

$O(N^4)$), aussi nous proposons de traiter un sous-ensemble S' représentatif, dont les éléments s sont choisis de façon aléatoire dans S .

5.5.3 Algorithme

Une façon simple d'obtenir un sous-ensemble S' est de choisir aléatoirement des couples de points (P_1, P_2) qui, lorsqu'ils sont reliés, définissent un segment. Un algorithme spécifique [Bre65] permet un calcul précis et rapide pour construire les segments. Cette méthode n'est pas optimale car elle introduit deux artéfacts :

1. Un effet de bord dû à la probabilité non uniforme d'un pixel à être traversé par un segment. Des approximations doivent être faites pour réduire le phénomène [Ber92].
2. Une répartition non équiprobable des directions des segments dans l'image [Dor84] [Kra91].

Les problèmes sont résolus si le tirage aléatoire d'un couple de pixels défini par $(x(P_1), y(P_1), x(P_2), y(P_2))$, est remplacé par le tirage aléatoire d'un pixel, d'une direction et d'une longueur définis par $(x(P_1), y(P_1), \theta, l)$. Ce quadruplet permet de calculer le point P_2 qui peut se situer ou non dans l'image. S'il est extérieur à l'image, seule la partie intérieure du segment est traitée.

Pour chaque segment s , un critère C peut être évalué sur les valeurs de $f(x)$ pour déterminer s'il passe par un contraste significatif afin de déduire s'il traverse ou non un contour. Le traitement effectué est itératif (algorithme 8). Chaque itération peut détecter un ou plusieurs éléments de contour.

La longueur du segment joue un rôle important sur l'amplitude de l'élément de contour trouvé. Les petits segments ont la capacité de détecter des éléments de contours faibles ou des discontinuités dues à la présence de texture (figure 63.b). Les segments de grande taille font ressortir les fortes discontinuités. Ainsi, une même région est traitée à différentes échelles. Le nombre de tirages effectués sur une image a été déterminé de façon empirique : nous avons choisi $Card(S')$ de l'ordre de N^2 . Pour une telle valeur, le processus est bien stabilisé visuellement et le temps de traitement reste raisonnable² (figure 64).

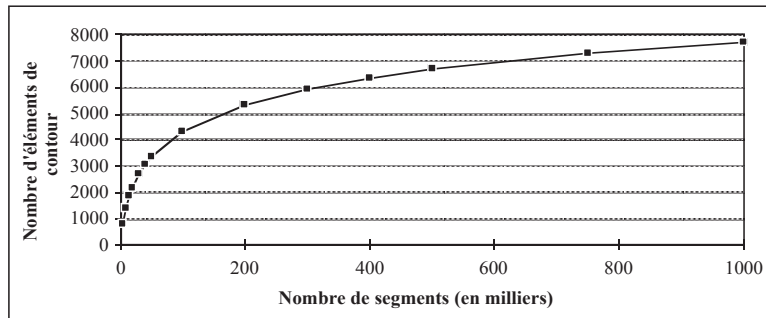


Fig. 64 - Nombre d'éléments de contours détectés en fonction du nombre de segments lancés

5.5.4 Critère de discontinuité adaptatif

Le critère C a été choisi de façon à être adaptatif à chaque segment. C peut être soit un critère unique, soit un critère multiple fondé sur des lois heuristiques ou statistiques. C ne doit pas inclure de seuil ou de paramètre statique afin que chaque segment conserve sa capacité d'auto-adaptation. L'algorithme recherche le (ou les) couple(s) de pixels s_m et s_{m+1} dont les niveaux de gris f_m et f_{m+1} forment le contraste maximal G_{Max} sur le segment :

$$G_{max} = \max_{i=0, \dots, l-2} (|f(i) \ominus f(i+1)|)$$

Le critère sélectionne les discontinuités significatives en utilisant m_f la moyenne et σ_f l'écart-type des contrastes de f : Si $G_{max} > m_f + \sigma_f$, alors s_m et s_{m+1} sont marqués sur la carte des éléments de contour.

²De l'ordre de 15 secondes pour une image 256×256 sur une station graphique SUN SPARCstation 5.

5.5.5 Prise en compte du bruit

Nous avons amélioré l'algorithme de base en le rendant plus robuste au bruit [Ber95b].

Le problème rencontré concerne les contrôles de cohérence à appliquer sur les éléments de contour détectés. Ce problème se pose systématiquement en segmentation d'image, quelles que soient les approches et les techniques utilisées. Il s'agit d'avoir des traitements peu sensibles au bruit et à la texture (détectée par les petits segments). Avec l'algorithme initial, notre méthode est très sensible au bruit (figure 66.a). Afin de ne conserver que des éléments de contour assez sûrs, nous utilisons l'information apportée par les segments voisins parallèles : lorsqu'un couple $(f(m), f(m+1))$ vérifie le critère C , les deux couples voisins parallèles (un de chaque côté, dans la direction de $[s_m, s_{m+1}]$) doivent générer un gradient de même signe que le couple $(f(m), f(m+1))$ (figure 65.a). Si oui, l'élément de contour potentiel est marqué sur la carte des contours (figure 65.a), sinon il est rejeté (figure 65.b). Les résultats sont ainsi nettement améliorés (figure 66.b).

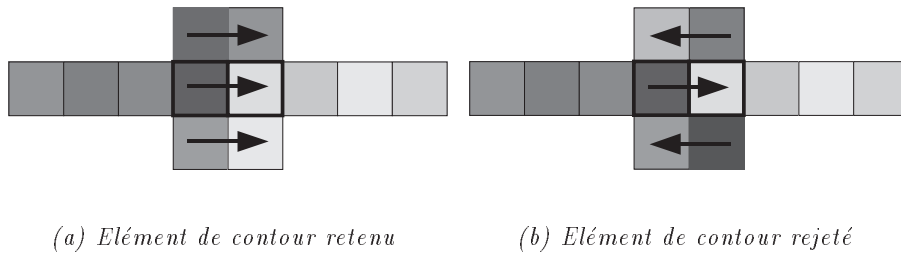


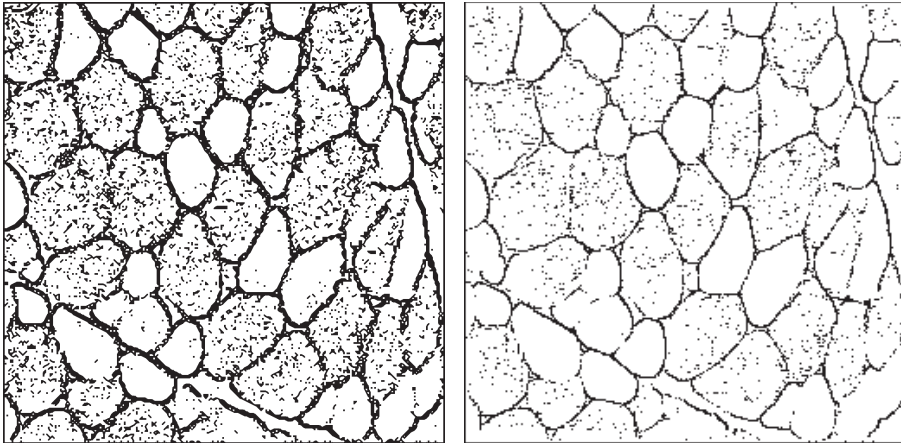
Fig. 65 - Prise en compte du bruit

La figure 67 montre les éléments de contour détectés pour l'image 'muscle' ainsi que la segmentation obtenue avec cette coopération. Des résultats comparatifs entre méthodes sont proposés au chapitre 7.

5.6 Application aux images de profondeur

Les images de profondeurs ou *range images* sont des images où le niveau de gris d'une entité est proportionnel à sa distance au capteur. Dans ce type d'image, on rencontre essentiellement des contours en escalier (objet en premier plan se découplant sur un arrière-plan éloigné) et des gradients de force variable (objet volumineux ayant une continuité en profondeur).

L'image utilisée pour les tests est donnée en figure 68 avec son histogramme des niveaux de gris et un profil caractéristique. La pyramide irrégulière classique est assez gênée par les gradients très étendus (figure 69). La coopération régions-contours résout en grande partie ce problème (figure 70).



(a) Les éléments de contours sont nombreux : 15315

(b) Les éléments de contour sont moins sensibles au bruit et moins nombreux : 5839

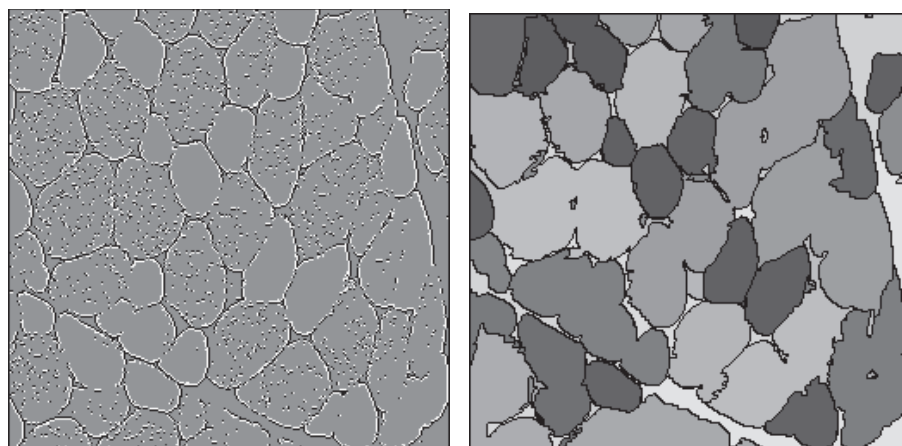
Fig. 66 - Détection stochastique et prise en compte du bruit. L'exemple montre des résultats obtenus avec 10^5 segments

Nous avons utilisé chacune des 2 méthodes avec 4 seuils globaux T_g différents afin de voir leur comportement. Le tableau 1 montre un comparatif des résultats obtenus. A l'aide de ces résultats, on peut noter essentiellement deux points :

Méthode	$T_g = 10$	$T_g = 20$	$T_g = 30$	$T_g = 40$	Moyenne	σ
non-coopérative	138	32	10	7	47	62
coopérative	25	24	17	12	20	6

Tab. 1 - Nombre de régions de l'apex

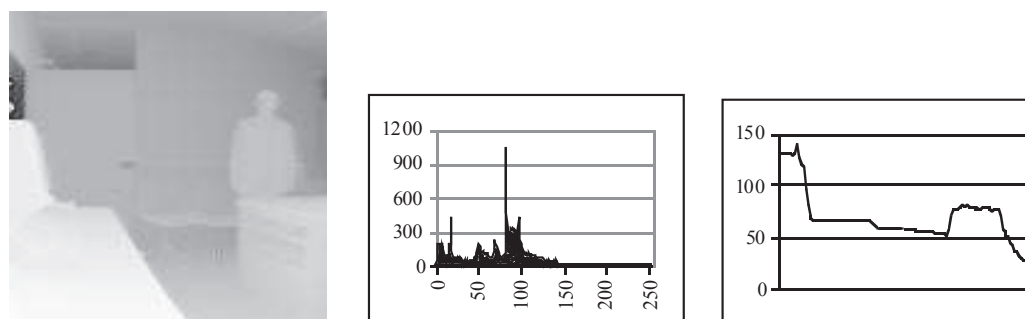
1. La méthode coopérative est la plus robuste en ce qui concerne le nombre de régions détectées en fonction du seuil utilisé. Comme l'image n'est pas beaucoup bruitée, la fusion imposée fonctionne bien pour traiter les dégradés ; la fusion empêchée permet de conserver le personnage. Pour T_g faible, la pyramide classique ne permet pas les fusions à l'intérieur d'un même dégradé, et en outre, de nombreuses petites régions en bordure des objets sont conservées.
2. En ce qui concerne la validité de la segmentation, la méthode coopérative est plus cohérente. Les entités significatives de l'image sont détectées, sans qu'il y ait de problèmes de dégradé. 2 fois sur 4, la porte est détectée malgré le



(a) Carte des éléments de contours
(chaque élément est composé d'un
pixel noir et d'un pixel blanc)

(b) Résultat de la segmentation

Fig. 67 - Coopération région-contour avec le détecteur de contours stochastique



(a) Image originale
(128 × 128)

(b) Histogramme des niveaux de gris

(c) Profil horizontal à mi-hauteur de l'image

Fig. 68 - Caractéristiques de l'image de profondeur utilisée

très faible contraste qui existe sur sa frontière (figure 68.c). En revanche, la pyramide classique offre des résultats où l'information significative n'est pas isolée ou absente.

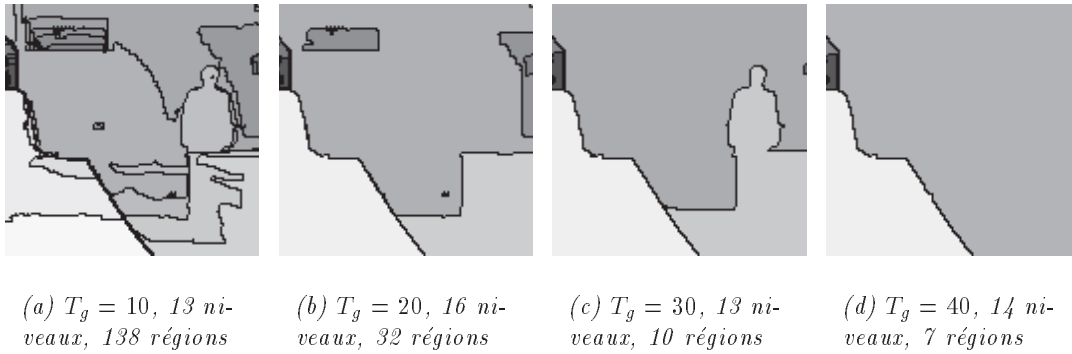


Fig. 69 - Segmentation sans coopération

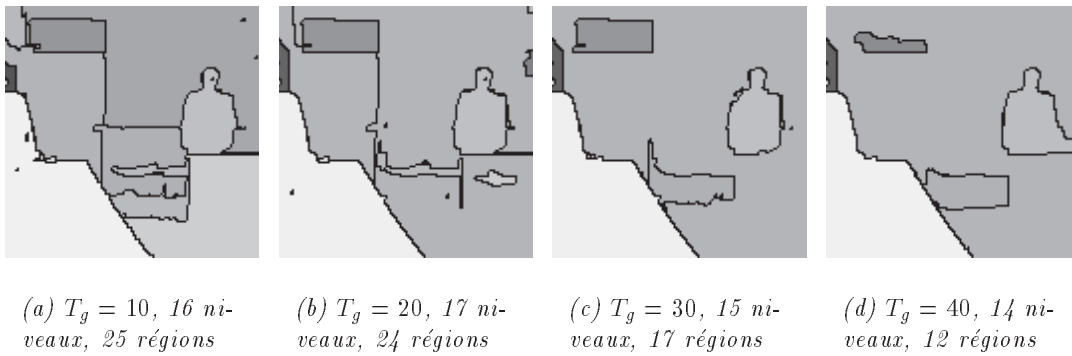


Fig. 70 - Segmentation avec coopération

5.7 Discussion et conclusion

D'après le comparatif du chapitre 7, la coopération région-contours améliore sensiblement les résultats de la segmentation. Il faudrait désormais mettre en place une gestion des 'fuites' en contrôlant les fusions qui s'opèrent vers les extrémités des contours : contrôles de cohérence, retardement des fusions, prise en compte de la topologie locale,

Le modèle proposé s'intègre bien dans la théorie de la pyramide et dans l'utilisation de la théorie des graphes. Les détecteurs de contour utilisés réduisent le rôle essentiel joué jusque-là par le seuil global de similarité : la structure pyramidale irrégulière devient globalement plus adaptative.

La souplesse de la coopération donne la possibilité d'utiliser les résultats fournis par la majorité des détecteurs.

Chapitre 6

Aspects multirésolution

Dans ce chapitre, nous nous intéressons à la segmentation multirésolution.

Dans un premier temps, nous montrons que la pyramide irrégulière classique n'est pas, telle quelle, une structure multirésolution. Néanmoins, elle comporte des éléments pouvant être utilisés pour la multirésolution.

Nous exploitons ses capacités en développant deux méthodes très distinctes :

La première utilise un paramètre d'échelle dont les différentes valeurs permettent d'effectuer sur une pyramide irrégulière déjà construite, des décompositions *top-down* de différentes résolutions.

Dans la seconde, nous présentons une approche originale dans laquelle nous construisons, lors d'un parcours *bottom-up* en plusieurs phases spécifiques, une pyramide d'images multirésolution.

6.1 Problématique

Cette section présente l'aspect multirésolution dans les méthodes de segmentation en général, et dans la méthode pyramidale irrégulière en particulier.

6.1.1 Terminologie

Le *Petit Larousse* définit la **résolution** en photocomposition et en PAO comme étant le nombre de points formés par le balayage d'un faisceau laser, sur une hauteur de 2,54 cm, pour assurer la restitution des textes et des images.

La résolution s'exprime donc en pixels par pouce. Selon que le rapport *pixel/pouce* soit plus ou moins grand, la résolution est plus ou moins élevée et les détails de l'image plus ou moins apparents.

Par extension, nous définissons la résolution comme la quantité de primitives qui décrivent une image. La **multirésolution** d'une image est donc sa représentation avec une quantité plus ou moins importante de primitives. En segmentation, les primitives correspondent à des régions.

6.1.2 Le cas des méthodes de segmentation

Notre système visuel étant capable d'analyser l'environnement à plusieurs résolutions, les méthodes de segmentation usuelles sont inadaptées de ce point de vue.

Les chercheurs, conscients de l'inadéquation des méthodes aux images, imputent ce problème aux traitements trop simplistes ou encore à la quantité trop faible d'information traitée.

Plutôt que de considérer l'image comme plusieurs problèmes réunis admettant une unique solution, nous préférons voir un problème qui admet plusieurs solutions.

Pour illustrer notre propos, remarquons que, face à une image à traiter, chaque méthode donne 'sa' meilleure segmentation, obtenue avec un paramétrage 'optimisé'. Pour un paramétrage différent (i.e. non optimisé), le résultat est généralement jugé 'moins bon', rarement 'aussi bon à une résolution différente'.

Dans cette recherche de la solution unique, nous considérons une région exacte que si elle est sémantiquement cohérente. Cette approche oblige à extraire les objets qui sont sémantiquement les plus importants. Dans la segmentation d'un visage, par exemple, il semble incohérent de ne pas extraire les composantes essentielles (nez, bouche, ...). En revanche, l'extraction de composantes moins essentielles (sourcils, fossettes, ...) est secondaire.

6.1.3 Le cas de la pyramide irrégulière

Avec la pyramide irrégulière, les régions d'un niveau donné ne sont plus contraintes par des critères géométriques. En revanche, ces régions ainsi que leur croissance sont

contraintes par la méthode de décimation locale employée. La localité de contraction du graphe (localité des décisions, de la décimation, du rattachement) induit un manque d'adaptativité des régions au contenu de l'image, cela pour la majorité des niveaux de la pyramide : dans les premiers niveaux, bien que le voisinage des régions et leur forme soient irréguliers, ils n'en sont pas moins aléatoires et sans contenu sémantique par rapport à l'image (figure 71).

Ainsi, les différentes segmentations des niveaux successifs de la pyramide ne constituent pas une représentation multirésolution d'une image. Néanmoins, la pyramide irrégulière, avec sa technique de croissance de régions en parallèle, offre d'autres moyens pour décrire une image à plusieurs résolutions.

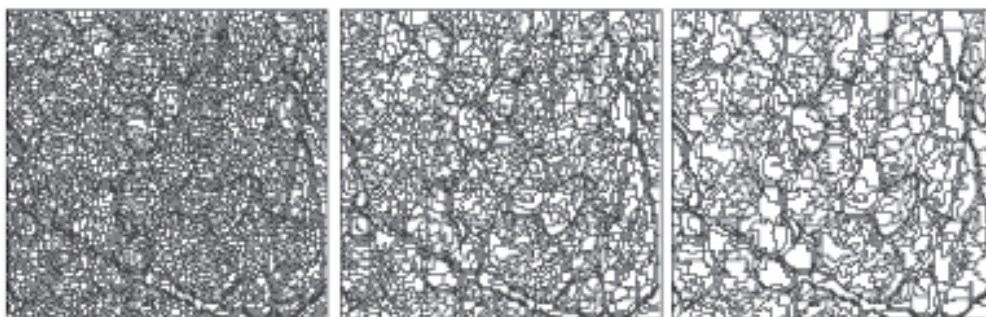


Fig. 71 - Contours des régions des niveaux 2, 3 et 4 de la pyramide irrégulière avec l'image 'muscle'. La présence de multirésolution est quasi inexistante

6.1.4 Éléments de multirésolution

Dans cette partie, nous allons discuter

- des attributs qui peuvent être importants pour mettre en évidence l'aspect multirésolution dans une image ;
- des potentiels de multirésolution de la pyramide irrégulière.

En segmentation, la notion de multirésolution est liée directement à la représentation ou non d'un objet de l'image originale.

La capacité de notre système visuel à distinguer un objet homogène en niveaux de gris¹ dans une scène (statique)² observée à une distance variable, dépend essentiellement de 4 critères classés par ordre d'importance décroissante :

1. la taille de l'objet par rapport au champ visuel ;
2. le contraste qu'il forme avec son voisinage ;

¹nous ne traitons pas ici du cas plus complexe de la texture, ni de phénomènes gestaltistes

²toute connaissance sémantique mise à part

3. son homogénéité propre ;
4. sa forme.

Un savant mélange de ces attributs fait qu'à une distance donnée d'une scène, nous sommes capables d'observer uniquement une certaine représentation (simplificatrice) de la réalité. Notre système visuel possède en effet une résolution maximale qui n'est pas infinie.

D'un point de vue segmentation d'image, l'existence ou la non existence d'une entité à une résolution donnée dépend donc des attributs cités.

Sur ce dernier point, nous pensons que la segmentation d'une image ne passe pas forcément par son partitionnement total. En effet, dans la plupart des cas, le résultat d'une segmentation est jugé par rapport aux connaissances sémantiques d'une image, et il est bien rare que toutes les zones d'une image aient la même importance ou même une signification intrinsèque. L'exemple typique est celui de quelques objets posés sur un fond uniforme. L'extraction seule des objets est significative et la région représentant le fond est auxiliaire et n'est d'ailleurs pas considérée comme une entité à part entière.

La pyramide irrégulière, nous l'avons dit, ne propose pas de segmentation multirésolution de l'image. Les partitionnements successifs peuvent être vus comme des partitionnements multirésolution, tout au plus, sans correspondance directe et bijective avec le contenu de l'image. Pour nous en convaincre, imaginons le traitement d'une image totalement uniforme. D'un côté, seule la taille de l'image originale nous renseigne sur sa résolution théorique (elle peut être indifféremment représentée en dimension 512×512 ou 128×128 , voire même de la taille d'un pixel), ce qui signifie que dans ce cas précis la multirésolution n'intervient pas. D'un autre côté, la pyramide nécessitera un certain nombre de niveaux pour segmenter l'image en une seule région. Tous les niveaux intermédiaires ne sont pas représentatifs du contenu de l'image, et sont formés de régions dont la forme est guidée uniquement par un processus stochastique. Cet exemple illustre bien la réponse insatisfaisante qu'apporte la pyramide irrégulière en segmentation multirésolution.

Néanmoins, nous remarquons que la croissance en parallèle de régions, apportée par la méthode, ne défavorise aucune partie de l'image, quel qu'en soit son contenu. Ainsi, chaque cellule de la pyramide évolue de niveau en niveau, avec une probabilité non nulle de passer par un état où elle correspond à une entité de l'image. Cependant, pour un seuil de similarité donné, toutes les entités ne sont pas extraites au même niveau, ce qui entraîne la disparition de certaines entités ou la non formation de certaines autres.

Pour conclure sur cette réflexion tournée vers les capacités de la pyramide irrégulière à exprimer la multirésolution, notons que grâce à cette structure, un phénomène d'émergence des entités contenues dans l'image lorsqu'on se rapproche de l'apex est observé. Nous désirons généraliser cette émergence à tous les niveaux, en régulant sa vitesse afin de parvenir à une représentation multirésolution du contenu de l'image.

6.2 Stratégie *top down*

En étudiant les différents niveaux d'une pyramide irrégulière, nous remarquons que les entités homogènes de l'image (ou régions significatives) sont souvent correctement segmentées ponctuellement avant de fusionner entre elles. Par extension, nous pouvons émettre l'hypothèse qu'une segmentation optimale³ (pour une résolution donnée) correspond à un partitionnement où chaque région serait choisie au niveau qui convient le mieux à sa représentation.

Partant d'une unique pyramide construite comme nous l'avons étudié dans le chapitre 4, nous désirons proposer, *a posteriori*, différentes segmentations correspondant à différentes résolutions en fonction d'un paramètre de résolution. Cette méthode est fondée sur un parcours *top down* de la pyramide. L'ensemble du traitement (construction ascendante puis parcours descendant) constitue ainsi une approche **fusion-division**.

6.2.1 Pyramide et arborescence

Jusqu'à présent, nous avons utilisé essentiellement la composante horizontale de la structure pyramidale : graphes d'adjacence, de similarité, sous-graphes. Il n'en est pas moins vrai que la pyramide est avant tout une structure verticale arborescente. L'arbre peut posséder une ou plusieurs racines, en fonction du nombre d'entités détectées dans l'image. La segmentation effectuée jusqu'alors est obtenue par construction de cet arbre lors d'un processus de type *bottom-up*.

6.2.2 Récursivité de la division irrégulière

Le chapitre 2 met en évidence les propriétés du découpage récursif dans les modèles de partitionnements géométriques et notamment dans le cas de la structure arborescente du *quadtrees*. Nous voulons utiliser l'analogie existant entre le découpage récursif dans ces modèles et les regroupements itératifs du modèle pyramidal irrégulier.

Les méthodes *top down* permettent de découper récursivement des régions selon un critère donné. Dans le cas de la pyramide irrégulière, le nombre de fils d'une cellule ainsi que leur forme sont variables. La construction *bottom up* de la pyramide permet de connaître la forme des tesselles du partitionnement à adopter, alors que dans le modèle *quadtrees* ce partitionnement est connu *a priori*. Dans notre méthode, un traitement *top down* de la pyramide traditionnelle va permettre d'affiner la segmentation (obtenue jusqu'alors à l'apex) et d'envisager une première approche multirésolution.

Le principe consiste à définir une nouvelle segmentation résultat qui sera formée de régions prises chacune dans le niveau de la pyramide qui convient au mieux à la

³nous parlons de segmentation optimale qualitativement, en termes de cohérence sémantique des régions extraites

finesse de détail qu'on désire localement (figure 72).

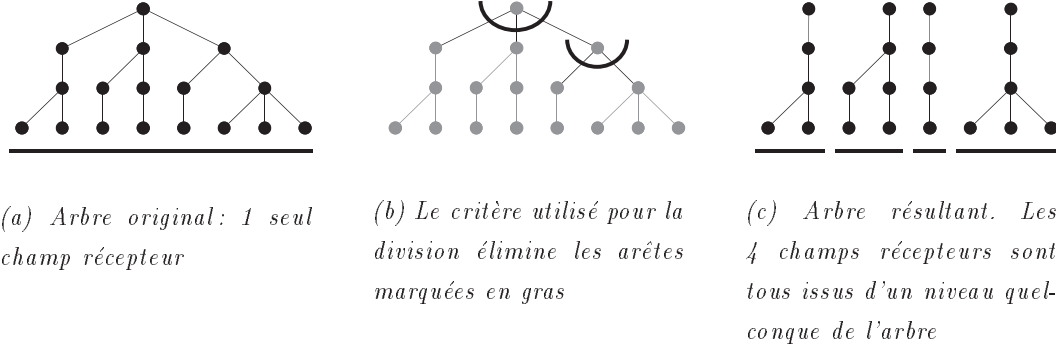


Fig. 72 - Division récursive des champs récepteurs

6.2.3 Traitement *Top-down* multirésolution

Le critère pour la division est différent du critère utilisé pour la fusion (comme dans l'approche multisegmentation proposée dans [Gag86]) :

Lors de la phase de construction *bottom up* de la pyramide, le critère de fusion utilisé est la similarité entre régions. Cette similarité est estimée entre les niveaux de gris moyens de régions adjacentes.

Dans la phase *top down*, nous proposons d'utiliser un critère fondé sur l'homogénéité. Celle-ci peut être évaluée sur l'écart-type des pixels formant le champ récepteur d'une région. Ainsi, de façon récursive, chaque région de niveau k ne satisfaisant pas le critère d'homogénéité est divisée suivant le partitionnement formé par ses fils de niveau $k \Leftrightarrow 1$.

Soit σ_M l'écart-type maximal toléré. Lorsque le traitement *top down* est achevé, toutes les régions qui forment le partitionnement obtenu ont un écart-type $\leq \sigma_M$.

Ce traitement *top down* est très rapide puisque la structure de l'arborescence est connue ainsi que la topologie et la morphologie des régions à chaque niveau. Si ce traitement est effectué plusieurs fois à partir de la même pyramide, pour différents seuils σ_M , les résultats obtenus possèdent les caractéristiques suivantes :

- Chaque grande entité très homogène apparaît comme une seule composante connexe même si σ_M est assez faible. La taille de ces régions est très peu dépendante de la valeur de σ_M .
- Les régions ont la forme d'entités optimalement homogènes.
- Le nombre de détails de l'image décroît lorsque σ_M croît et inversement.

- La résolution dépend d'un seul paramètre : σ_M qu'on peut rapprocher du **paramètre d'échelle** de l'approche *scale-space* (§ 2.2.4).

Dans cette optique, le calcul de l'écart-type doit être effectué une unique fois, pour chaque région, lors de la construction de la pyramide. La formule incrémentale utilisée est donnée au § 4.2.8.

σ_M peut être considéré comme un paramètre d'échelle possible pour la représentation multirésolution d'une image. Son utilisation est intéressante parce que la notion d'écart-type d'une population fait intervenir fortement la notion de taille de la population considérée. Comme nous l'avons déjà dit, l'existence d'une entité à une résolution donnée est largement dépendante de sa taille.

6.2.4 Paramètres d'échelle

L'écart-type utilisé comme paramètre d'échelle est valable pour une certaine catégorie d'images qui sont peu texturées et peu bruitées. On peut envisager d'autres paramètres d'échelle en fonction du type d'image utilisée et des connaissances sémantiques que l'on veut extraire *a priori*. Le **facteur de forme** f_f est un attribut qui permet de caractériser⁴ la forme d'un objet. Dans notre cas, il peut permettre d'extraire des régions dont la morphologie est plus ou moins simple. Pour une région s , il est donné par :

$$f_f = \frac{4\pi S(s)}{P(s)^2}$$

où $S(s)$ et $P(s)$ sont respectivement la surface et le périmètre de la région. La formule fournit une valeur normalisée : $f_f \in [0, 1]$. Les valeurs limites correspondent au cercle ($f_f = 1$) et au segment ($f_f = 0$).

Pour extraire les objets les plus contrastés, on peut choisir un paramètre d'échelle c_M prenant en compte le contraste moyen $\overline{c(s)}$ d'une région par rapport à ses voisines :

$$\overline{c(s)} = |g(s) \Leftrightarrow \frac{\sum_i g(V_i(s))}{i}|$$

Le contraste moyen peut être pondéré par la longueur de la frontière commune entre la région et chacune de ses voisines.

Sur certaines images, un paramètre d'échelle fondé sur la taille des régions peut être utilisé. Ce paramètre a tendance néanmoins à fournir des résultats qui se rapprochent des segmentations fournies par les niveaux bas de la pyramide lorsque la résolution augmente. Enfin, on peut envisager des paramètres d'échelle prenant en compte plusieurs des attributs cités ci-dessus.

⁴de façon quantitative

6.2.5 Nouvelle structure de données

Pour évaluer *a posteriori* un critère sur une région de niveau k et connaître son partitionnement en régions de niveau $k \Leftrightarrow 1$, il faut disposer des informations relatives à ces deux niveaux. Par récursivité, il faut connaître tous les niveaux de la pyramide.

Dans la pyramide irrégulière classique, chaque composante connexe d'une région peut être reconstituée grâce au lien qui relie chaque pixel à son ascendant du niveau en construction. Afin de pouvoir reconstituer une région quelconque d'un niveau quelconque, il faut conserver pour chaque pixel autant de liens vers ses ascendants successifs qu'il y a de niveaux dans la pyramide (figure 73).

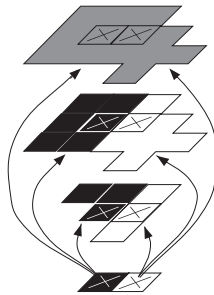


Fig. 73 - Deux pixels et leurs ascendants successifs : exemple. Les flèches représentent les liens de champs récepteurs

Nous avons modifié la structure de données de la façon suivante :

- Chaque sommet de niveau k possède des liens avec ses fils de niveau $k \Leftrightarrow 1$, afin de pouvoir effectuer le traitement descendant de division.
- Chaque pixel de l'image possède un lien avec le sommet qui est son ascendant à chacun des niveaux. Cette liste d'ascendants permet de reconstituer le champ récepteur d'un sommet pour un niveau quelconque.

Cette structure nous permet d'effectuer le traitement *top down* de deux manières qui donnent des résultats identiques : l'une récursive (algorithme 9), l'autre itérative (algorithme 10).

- Le traitement récursif permet de regrouper tous les pixels représentés par un sommet d'un niveau donné, c'est-à-dire de définir la région qui satisfait le critère utilisé. La méthode est coûteuse en temps mais ne nécessite la mémorisation que des différents niveaux de la pyramide.
- Le traitement itératif cherche pour chaque pixel son ascendant de niveau le plus élevé satisfaisant le critère choisi. Cette technique permet d'obtenir des résultats en temps réel mais corollairement, elle est coûteuse en place mémoire.

Par exemple, pour mémoriser les liens de chaque pixel avec tous ses ascendants successifs dans une pyramide de 20 niveaux d'une image 256×256 , la place mémoire requise est :

$$65536 \text{ pixels} \times 20 \text{ niveaux} \times 4 \text{ octets} = 5 \text{ mégaoctets.}$$

Algorithme 9 (*Parcours top-down récursif*)

```
/* divise en fils un sommet donné d'un niveau donné */

procédure DIVISER(sommet, niveau)
début
  pour chaque fils f_i de sommet faire
  début
    si (ecart-type(f_i) > ecart_type_toleré)
      alors DIVISER(f_i, niveau - 1);
    sinon
      pour chaque pixel p_j dont f_i est un ascendant
        représentant(p_j) = f_i;
  fin;
fin.
```

Algorithme 10 (*Parcours top-down itératif*)

```
/* recherche pour chaque pixel le champ récepteur qui vérifie le critère */

début
  pour chaque pixel p_j faire
  début
    représentant(p_j) = NULL;
    niveau <- apex;
    tant que (représentant(p_j) = NULL) faire
    début
      si (ecart-type(champ_récepteur[niveau] <= ecart_type_toleré)
        alors représentant(p_j) <- champ_récepteur[niveau];
      sinon
        niveau <- niveau - 1;
    fin;
  fin;
fin.
```


6.2.6 Résultats expérimentaux

Pour illustrer les résultats obtenus avec la méthode, nous utilisons une scène d'extérieur qui se prête à de nombreuses résolutions. L'approche (de type *fusion - division*) est assez simple à mettre en œuvre et offre des résultats satisfaisants si on considère son extrême simplicité et ses temps de calcul réduits. Ces résultats sont intéressants non seulement du point de vue multirésolution, mais en outre, la méthode permet à l'utilisateur, à partir d'une seule pyramide irrégulière, d'affiner sa segmentation par la recherche en temps réel de la résolution la mieux adaptée à son besoin (figure 74).

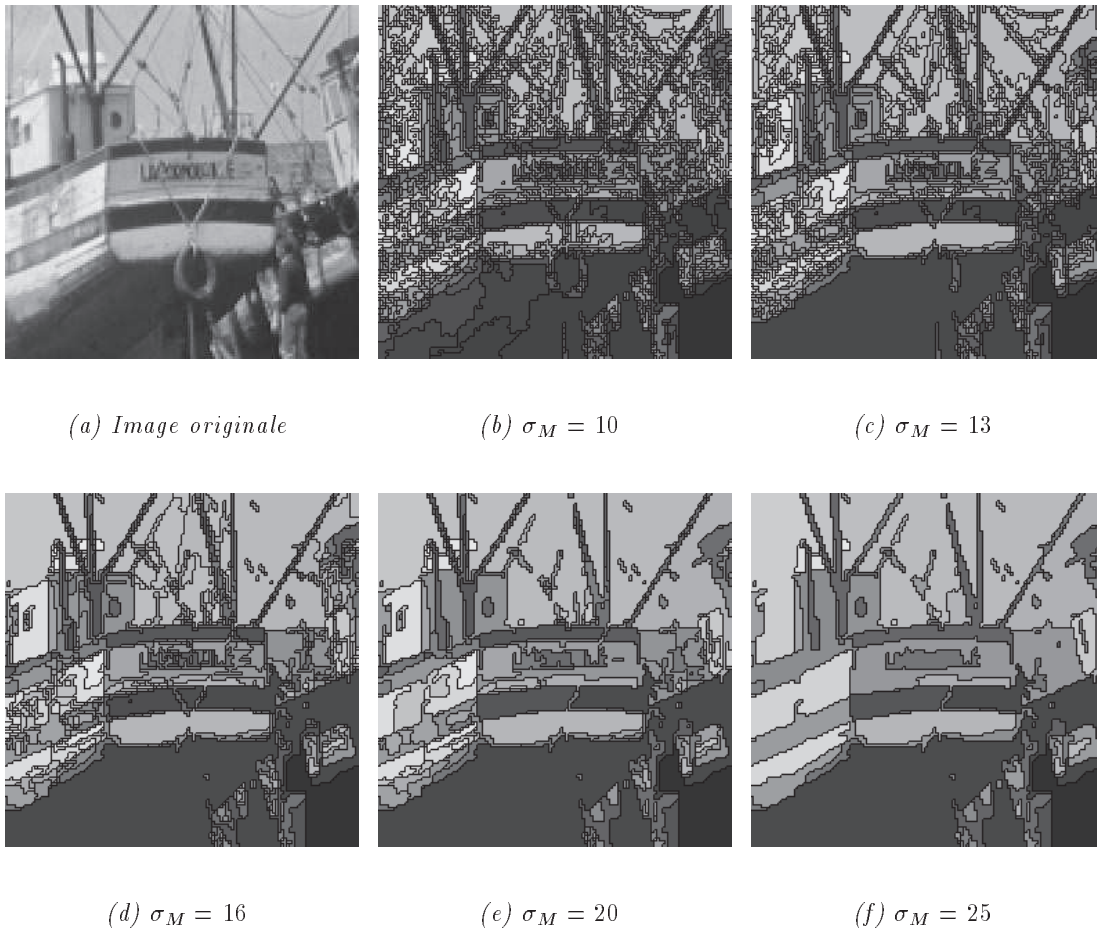


Fig. 74 - Pyramide multirésolution obtenue par variation du paramètre d'échelle 'écart-type'

6.3 Pyramide multirésolution

L'approche étudiée dans la section précédente est à la fois facile à mettre en œuvre, rapide et souple d'emploi. Les résultats obtenus apportent des premiers éléments de multirésolution.

Dans cette section, nous étudions comment la pyramide irrégulière classique peut être contrainte pour obtenir directement, lors de sa construction, des segmentations à plusieurs résolutions.

Cette méthode est plus originale et plus complexe que la précédente dans le sens où elle tente d'apporter une réponse au problème de cohérence des partitionnements posé par la contraction locale des graphes.

6.3.1 Décimation d'ordre étendu

L'algorithme de construction de la pyramide irrégulière est parallèle. Ce parallélisme pose un problème majeur par rapport à la multirésolution : la croissance de chaque région se fait localement, donc de façon assez uniforme sur toute l'image. Cette constatation, tout à fait satisfaisante pour la complexité algorithmique et le parallélisme, empêche des vitesses de croissance différentes pour des zones de l'image plus ou moins homogènes. En effet, un sommet ne peut ponctuellement fusionner qu'avec les sommets dont il est directement voisin : les algorithmes de décimation utilisés fonctionnent sur des voisinages d'ordre⁵ 1.

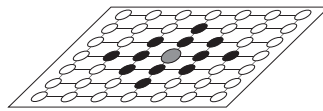


Fig. 75 - Les sommets noirs définissent le voisinage d'ordre 2 du sommet central

Malgré l'existence d'une similarité parfaite entre de nombreux voisins, leur fusion complète ne sera achevée qu'après un certain nombre de décimations, c'est-à-dire après la construction de plusieurs niveaux.

Une solution permettant des croissances de régions rapides réside dans une décimation adaptée au contexte, non limitée au voisinage immédiat, et appelée **décimation d'ordre étendu** (ou décimation étendue). La remise en cause de la décimation locale existante, dans le but d'obtenir cette nouvelle décimation, soulève trois problèmes :

1. déterminer un critère permettant d'extraire d'un graphe G , les sous-graphes J_i sur lesquels effectuer la décimation étendue ;

⁵Le voisinage d'ordre n d'un sommet s est l'ensemble des sommets $V(s)$ dont chaque sommet s_i est relié à s par un chemin de longueur l , $0 < l \leq n$ (figure 75). Le voisinage d'ordre 1 (utilisé jusqu'à présent) correspond au voisinage immédiat.

2. déterminer un opérateur Op qui permet de décimer chaque J_i , de façon à n'en extraire qu'un seul sommet ;
3. contracter correctement le graphe en respectant sa topologie.

Dans le cas de la décimation locale (dont les règles sont données au § 4.2.5), J_i est une étoile et Op est la maximisation d'un attribut stochastique ou déterministe. La topologie, quant à elle, est respectée par mise à jour des relations de voisinage immédiat.

Critère d'appartenance à un sous-graphe

Avant d'effectuer la décimation, il est nécessaire de déterminer un ensemble de sous-graphes J_i dont chacun optimise un critère donné. Notre but étant de décimer plus rapidement les zones homogènes, nous avons choisi un **critère d'unification** fondé sur l'homogénéité locale⁶ :

$$hom(s) = \begin{cases} vrai & \text{si } \forall s_l \in V(s), sim(s, s_l) \text{ est vrai} \\ 0 & \text{sinon} \end{cases}$$

Le prédicat $sim(s_i, s_j)$ est défini simplement par :

$$sim(s_i, s_j) = \begin{cases} vrai & \text{si } |g(s_i) \leftrightarrow g(s_j)| < T'_g \\ 0 & \text{sinon} \end{cases}$$

L'homogénéité d'une zone doit être adaptative localement et non pas globalement. Par exemple, les pixels appartenant à un dégradé sont localement similaires entre eux alors qu'entre le début et la fin du dégradé, la différence de niveau de gris peut être importante par rapport au seul global de similarité fixé (similarité non transitive).

Dans ce but, chaque pixel évalue le critère d'homogénéité et établit s'il est homogène ou non par rapport à son voisinage. La décision est locale, et l'étiquette de la zone homogène à laquelle appartient le pixel est *a priori* non connue. Pour assurer l'homogénéité totale à l'intérieur d'un sous-graphe, nous avons choisi un critère d'appartenance strict qui évite les fusions potentiellement fausses et qui n'anticipe pas la possible évolution des similarités sur les niveaux supérieurs de la pyramide.

Un pixel appartient à un sous-graphe J_i s'il est homogène par rapport à son voisinage immédiat :

$$s \in J_i \text{ ssi } hom(s)$$

Pour les sous-graphes ne comprenant qu'un seul sommet homogène, la décimation d'ordre 1 est naturellement utilisée. La décimation étendue, quant à elle, est réservée aux sous-graphes comprenant plusieurs sommets homogènes :

⁶critère évalué localement, parallélisable

$$s \in J_i \text{ ssi } \text{hom}(s) \text{ et } \exists s_h \in V(s), \text{hom}(s_h)$$

Dans un sous-graphe J_i , chaque sommet est similaire selon le prédicat donné à tous ses voisins sur G . Ce critère induit un effet de bord : tous les pixels périphériques à un sous-graphe sont similaires localement aux pixels de J_i mais ne possèdent pas un voisinage homogène (figure 76).

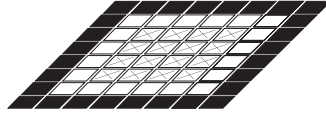


Fig. 76 - Effet de bord du critère d'unification. Seuls les pixels de la région blanche marqués d'une croix ont un voisinage homogène

Par mesure de simplicité et compte tenu de la caractéristique élémentaire des J_i , nous appellerons ces sous-graphes des **graphes homogènes**. Ces sous-graphes pourraient également être appelés **graphes d'équivalence** : la relation qui permet la connexion de leur sommet est réflexive, symétrique et transitive.

Le regroupement des sommets adjacents homogènes permet l'émergence d'une région homogène.

Remarque : l'ensemble des graphes homogènes (appelés **noyaux de contraction** dans un contexte plus global [Kro95]) pourrait être obtenu à partir d'un arbre couvrant minimal de G , comme il est proposé dans [Kro95] : le découpage en sous-arbres serait effectué sur les arêtes dont le poids est supérieur à un seuil de similarité donné. Chaque sous-arbre constituerait une zone homogène à part entière.

Extraction des sous-graphes

Nous posons ici le problème de l'extraction des graphes homogènes, c'est-à-dire la capacité à retrouver tous les sommets appartenant à ce sous-graphe.

Comme l'évaluation de ce critère est locale, chaque sommet est étiqueté comme étant homogène ou non, sans distinction d'appartenance à un graphe homogène plutôt qu'à un autre. La recherche de l'ensemble des sommets d'un graphe homogène peut s'effectuer à partir d'un sommet homogène quelconque s_i (i.e. vérifiant le prédicat $\text{hom}(s_i)$). L'extraction complète correspond ensuite à étiqueter la composante connexe des sommets s_i vérifiant $\text{hom}(s_i)$. Un certain nombre de méthodes peuvent être trouvées dans la littérature. Notons un algorithme séquentiel en deux passages [Ros66] et un algorithme parallèle [Cha85]. Les différents sous-graphes ne peuvent pas être adjacents 2 à 2 car il existe toujours au moins un chemin de longueur ≥ 3 qui relie 2 sous-graphes :

- 2 sommets homogènes s_i et s_k adjacents appartiennent au même sous-graphe, par définition de l'homogénéité :

- Soit un chemin de longueur 2 entre s_i et s_k tel que (s_i, s_j) et $(s_j, s_k) \in A$.

$$hom(s_i) \Rightarrow sim(s_i, s_j)$$

$$hom(s_k) \Rightarrow sim(s_k, s_j)$$

$$sim(s_i, s_j) \text{ et } sim(s_k, s_j) \Leftrightarrow hom(s_j)$$

Un sommet reliant 2 sommets homogènes est lui-aussi homogène. Ces 3 sommets appartiennent au même graphe homogène.

Décimation des sous-graphes

Le problème posé est le suivant : pour chaque sous-graphe, il ne faut retenir qu'un seul sommet survivant.

Pour conserver l'aspect parallélisable de la méthode, il faut effectuer la recherche d'un sommet survivant par la construction d'une pyramide à base locale [Kon93] dont l'apex ne sera constitué que d'un sommet (figure 77). Dans ce cas, un pro-

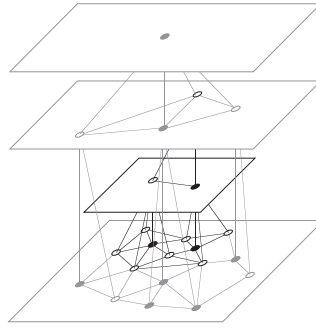


Fig. 77 - Pyramide de graphes comportant une pyramide à base locale (en noir)

blème d'implantation apparaît : il faut effectuer récursivement la construction d'une pyramide irrégulière. Nous n'avons pas retenu cette solution qui d'une part, demande une restructuration de la pyramide et d'autre part, apporte des temps de traitement élevés. Nous avons développé une technique séquentielle qui nous a permis de conserver à la fois la structure actuelle de données ainsi que les différentes phases du traitement utilisées jusque là. La décimation s'opère de la façon suivante :

1. Une première décimation stochastique permet de faire une sélection des sommets survivants, comme précédemment.
2. Le premier sommet traité survivant d'un graphe homogène J_i est choisi comme seul survivant définitif s_i chargé de représenter le graphe J_i au niveau supérieur. s_i communique à tous ses voisins dans J_i qu'il est le **sommet unificateur** du graphe. A leur tour, chacun de ces voisins propagera le sommet unificateur à l'ensemble du sous-graphe.

Nous avons choisi comme sommet unificateur le premier sommet survivant rencontré dans le traitement séquentiel. En effet, le choix du sommet n'importe pas puisque les sommets non-survivants du graphe homogène n'ont d'autre alternative que de fusionner avec l'unique sommet unificateur de J_i .

La propagation du sommet unificateur est accompagnée de traitements annexes assurant une mise à jour du graphe d'adjacence ainsi qu'un contrôle de la connexité du graphe homogène traité :

- une arête d'adjacence est rajoutée entre chaque sommet de J_i et le sommet unificateur s_i ;
- chaque sommet non survivant $\notin J_i$ et adjacent à J_i qui ne possède qu'un voisin survivant $\in J_i$ est forcé à être survivant.

Comme le traitement est séquentiel, il arrive fréquemment que plusieurs sommets unificateurs se propagent en même temps pour représenter une même zone homogène (figure 79).

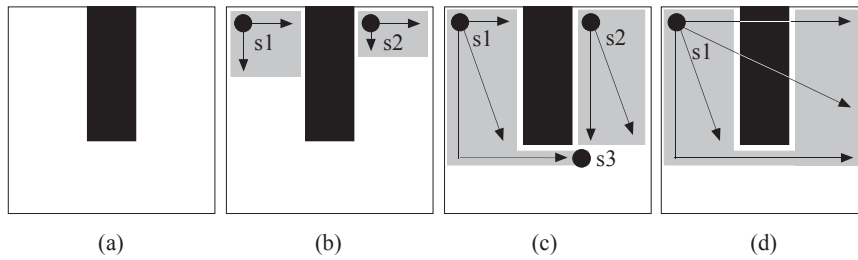


Fig. 78 - Propagation des sommets unificateurs. (a) montre une image avec une zone très homogène blanche. L'image est parcourue en séquence et un premier sommet unificateur s_1 est sélectionné et propagé (b). Un second sommet s_2 est sélectionné aussi et se propage 'en parallèle'. Les deux propagations se rencontrent en s_3 (c). Un seul sommet survit et étend ses propagations (d)

Le phénomène se détecte aisément lorsque les deux propagations se rencontrent, c'est-à-dire lorsqu'un sommet homogène veut propager son sommet unificateur à un voisin qui possède déjà un sommet unificateur différent (figure 78). Dans ce cas, un seul sommet unificateur est conservé et tout le voisinage du sommet non conservé est transféré au voisinage du sommet unificateur qui subsiste. Comme les voisinages transférés peuvent être importants, c'est le sommet unificateur de degré le plus faible qui est éliminé.

Le partitionnement obtenu au premier niveau montre clairement l'impact de la décimation étendue sur l'émergence d'entités homogènes. (figure 79).

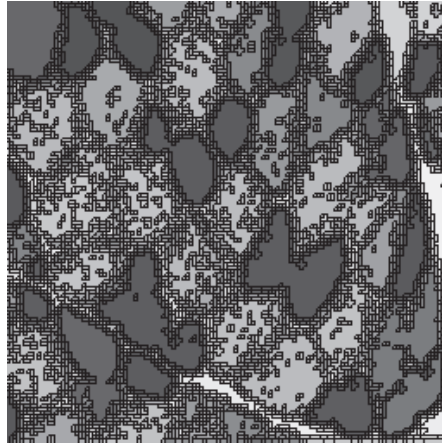


Fig. 79 - Niveau 1. 5053 régions. La croissance de régions est très rapide dans les zones homogènes. L'image utilisée est de taille 128×128

6.3.2 Extraction de composantes

Avec la décimation étendue, nous ajoutons à la croissance de régions en parallèle (c'est-à-dire locale) une croissance de régions à part entière qui est d'autant plus efficace que l'image est constituée de zones homogènes. Alors que précédemment, la taille des régions du premier niveau était limitée à 5 ou 9 pixels (respectivement pour la 4 ou la 8-connexité), les régions peuvent désormais avoir une taille beaucoup plus importante. En fonction du seuil T'_g utilisé, on remarque l'émergence d'entités dès le premier niveau. En déterminant une taille limite T_t et en conservant toutes les régions dont la taille, exprimée en pixels, est supérieure à T_t , on obtient une carte qui caractérise de façon plus ou moins grossière les objets de l'image (figure 80.a). Le nombre d'entités détectées dépend de T'_g et de T_t :

- T'_g permet de déterminer la dissemblance maximale entre sommets pour caractériser l'homogénéité choisie et donc déterminer une fiabilité plus ou moins grande dans la cohérence des objets.
- T_t permet de déterminer la taille à partir de laquelle on considère une zone homogène comme représentant une entité propre de l'image [Fal94]. Ce seuil permet de faire abstraction des petites régions qui, par leur taille, ne correspondent pas à des entités importantes.

Pour obtenir une extraction des composantes de l'image, il est préférable, dans la majorité des cas, de connaître le nombre approximatif n d'objets dans l'image, plutôt que de déterminer une taille minimale T_t . La correspondance peut alors être établie entre les 2 valeurs. Si $\text{Card}(0) = N^2$, une segmentation donnée par le niveau

k de la pyramide peut s'exprimer quantitativement :

$$\text{Card}(k) = \sum_{i=1}^{N^2} \alpha_i$$

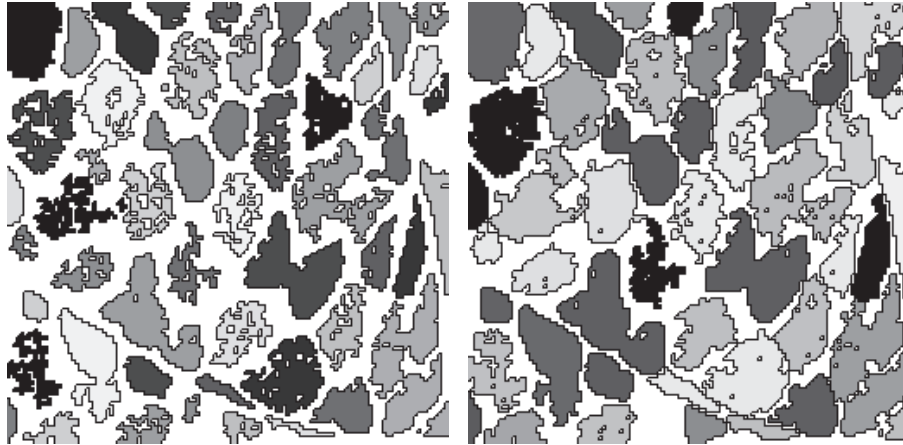
où α_i ($0 \leq \alpha_i \leq N^2$) est le nombre de régions de taille i qui forment une partition de l'image :

$$\sum_{i=1}^{N^2} \alpha_i i = N^2$$

En partant de $j = N^2$, il est facile de trouver, par cumul des α_j :

$$n = \sum_{j=t}^{N^2} \alpha_j$$

tel que t correspond à la taille limite recherchée T_t .



(a) Extraction effectuée au niveau 1

(b) Extraction effectuée au niveau 2

Fig. 80 - Extraction des objets (en fausses couleurs) de la figure précédente. Seuls sont extraits les 50 objets de plus grande surface

Nous avons vu en § 6.3.1 l'effet de bord engendré par le critère d'unification utilisé. Afin d'extraire la totalité de chaque composante connexe, il faut les sélectionner au second niveau et non plus au premier. Ce choix est important puisque la sélection des composantes est effectuée sur leur taille. Les objets obtenus sont alors plus compacts (figure 80.b).

6.3.3 De la pyramide stochastique à la pyramide adaptative

La décimation étendue a été effectuée à l'aide de la décimation stochastique. L'utilisation d'un traitement stochastique soulève un problème : la probabilité pour un sommet d'être sélectionné pour survivre au niveau supérieur est liée implicitement à son nombre de voisins. Cependant, ce critère quantitatif ne prend pas d'autres informations en ligne de compte. Ce mode de fonctionnement défavorise les petites régions qui se retrouvent enfermées au cœur de régions plus importantes : les premières ont un nombre faible de voisins (souvent un seul), alors que les dernières ont un nombre important de voisins puisqu'elles sont entourées elles-mêmes par d'autres régions. La loi de probabilité veut que les régions encerclées ne fusionnent que lorsque les régions encerclantes ne peuvent plus croître vers l'extérieur. Ces fusions ne s'opèrent que vers l'apex de la pyramide (figure 81).

Pour palier ce problème, nous proposons d'utiliser la pyramide adaptative (§ 4.2.5) munie d'un opérateur d'intérêt adéquat. Rappelons que la décimation retient dans ce cas les sommets qui optimisent localement cet opérateur d'intérêt. Nous avons testé 2 opérateurs :

- un qui prend en compte le degré de chaque sommet et qui permet aux sommets ayant le plus de voisins de survivre ;
- un autre qui prend en compte la taille des régions afin que les plus petites fusionnent aux plus grandes.

L'utilisation d'un tel opérateur d'intérêt résout le problème posé (figure 82).

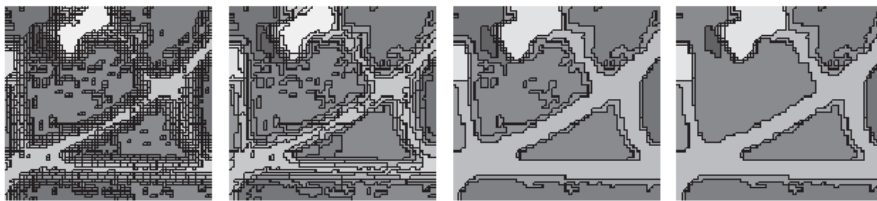


Fig. 81 - Problème posé par la décimation stochastique. Les petites régions encerclées fusionnent tardivement et les taux de décimation successifs sont faibles. De gauche à droite: niveau 1: 1305 régions, niveau 3: 316 régions, niveau 14: 98 régions, niveau 15: 85 régions

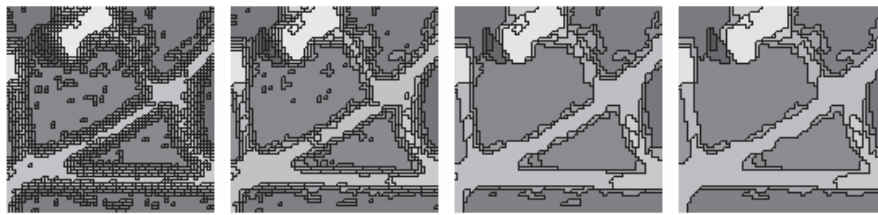


Fig. 82 - Problème résolu par la décimation adaptative. La pyramide converge vite. De gauche à droite : niveau 1 : 1348 régions, niveau 2 : 413 régions, niveau 3 : 212 régions, niveau 7 : 116 régions

En même temps, on remarque que les taux de décimation sont beaucoup plus importants. Ceci est normal : sur la pyramide obtenue avec la décimation d'ordre étendu, on obtient dès le premier niveau 2 types de régions :

1. De grosses régions comportant plusieurs dizaines (voir plusieurs centaines ou milliers de pixels) qui sont peu nombreuses et qui possèdent un nombre élevé de voisins (du fait de leur périmètre important). Leur capacité à récupérer des petites régions voisines est élevée.
2. De petites régions qui n'ont pas profité de la décimation étendue, qui ont donc une petite taille (moins d'une dizaine de pixels et souvent 2 ou 3 pixels seulement), et qui possèdent très peu de voisins. Ces régions sont en périphérie des grosses régions et sont rapidement conquises.

Les expérimentations effectuées ne permettent pas de déterminer si l'utilisation d'un attribut plutôt que d'un autre donne de meilleurs résultats : les 2 informations sont fortement corrélées.

6.3.4 Enrichissement des composantes

La décimation étendue et l'extraction de composantes nous ont permis d'obtenir l'information pertinente de l'image. Néanmoins, le partitionnement de l'image n'est pas terminé.

Germes et racines

Lorsque l'information récupérée lors de l'extraction des composantes du premier (ou second) niveau semble assez pertinente, nous proposons de l'utiliser comme **germes** de la croissance de régions et comme **racines** de la pyramide.

- Par sa qualité de germe, chacune de ces régions continuera sa croissance en annexant itérativement toutes les régions qui l'entourent. Cette croissance est

effectuée localement sur toute l'image donc sans défavoriser *a priori* une région particulière.

- Si chacune des régions extraites est considérée comme une racine, elle figurera systématiquement comme une région à part entière à l'apex de la pyramide. D'une part, ceci évite à deux entités extraites de fusionner. D'autre part, on peut construire la pyramide jusqu'à obtenir uniquement le nombre de régions voulu, c'est-à-dire les racines.

Croissance des germes

La croissance des germes ne peut s'effectuer qu'à condition de les forcer à survivre sur tous les niveaux supérieurs, lors du processus de décimation. Pour parvenir à fusionner toutes les régions qui ne sont pas des germes, et ainsi pour ne conserver que des racines à l'apex, il faut relâcher le critère de similarité et ne conserver que le critère de similarité local. Ce dernier permet aux fusions de s'effectuer dans l'ordre qui respecte le plus les homogénéités. La figure 83 montre le résultat obtenu sur l'image 'muscle'. L'approche correspond non seulement à une croissance de germes mais également à une croissance des régions non germes, en fonction de la topologie locale. Ces dernières ont la possibilité de croître avant d'être finalement absorbées par les germes.

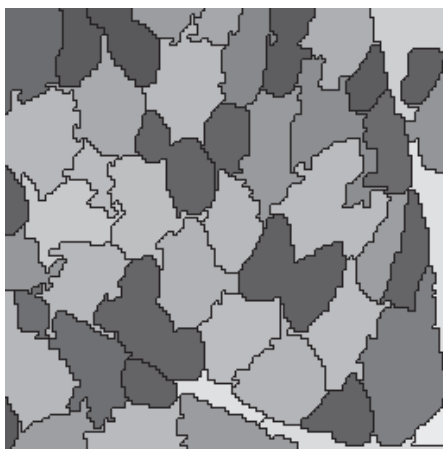


Fig. 83 - Segmentation comportant uniquement les racines extraites au premier niveau (50 régions)

6.3.5 Vers une pyramide d'images multirésolution

L'utilisation conjointe de la décimation étendue, de l'extraction des germes et de leur croissance par relâchement du critère de similarité induit la construction

d'une pyramide irrégulière d'un nouveau type. Chaque niveau de la pyramide est une segmentation à part entière. La taille des régions est régulée automatiquement par l'utilisation d'un graphe de similarité orienté. Même les zones qui contrastent fortement avec leurs voisines sont amenées à être absorbées si leur taille initiale n'est pas suffisante pour qu'elles soient des germes. La haute résolution n'empêche pas aux zones homogènes de constituer des entités à part entière, comme c'est le cas pour notre système visuel.

Les figures 84.a et b montrent les différences essentielles qui existent, au niveau des résultats, entre l'approche traditionnelle et la nôtre. La hauteur de la pyramide est en moyenne réduite de moitié grâce à l'utilisation de la décimation étendue et au relâchement du critère de similarité, qui donne des rapports de contraction élevés.

6.3.6 Résultats expérimentaux

La figure 85 permet de comparer la multirésolution obtenue par la méthode avec les résultats obtenus par la méthode *top-down* (figure 74, § 6.2.6).

La méthode, nettement orientée description multirésolution, donne également des résultats intéressants en segmentation. Des résultats comparatifs sont proposés au chapitre 7. Il nous paraît important de dégager les points forts et les points faibles de la méthode.

Tout d'abord les points forts ou tout au moins positifs :

- Émergence rapide des entités (plus de contrainte de taille de régions).
- Réduction du nombre de niveaux de la pyramide accompagnée d'une diminution des temps de traitement.
- Multirésolution réelle : l'information est significative à tous les niveaux de la pyramide.
- Abandon des traitements particuliers relatifs aux petites régions et à la troncature de la pyramide.
- La possibilité de donner le nombre de régions désiré pour la segmentation de résolution minimale.
- La croissance de région permet de délimiter des régions qui sont séparées par une région très fine et peu contrastée comme on peut le voir sur les haubans du chalutier de la figure 85.

Face à ces caractéristiques encourageantes, notons les faiblesses de la méthode :

- Le seuil T'_g influe directement sur la sélection des germes. La méthode est sensible à la variation de ce seuil, de la même manière que la méthode traditionnelle est sensible à T_g .

- Les formes filiformes (allongées et minces) sont très défavorisées par le critère d'homogénéité.
- La méthode donne de bons résultats avec des images peu bruitées, mais une sur-segmentation dès que l'image est bruitée. Dans ce cas, un filtrage de l'image est nécessaire.
- La décimation étendue ne peut pas être utilisée de façon efficace à tous les niveaux de la pyramide. Seule la construction du premier niveau en bénéficie.
- La croissance de région peut être anarchique lorsque les germes sont peu fiables.

6.4 Le cas des images texturées

Les images texturées sont particulières dans le sens où les techniques de segmentation traditionnelles sont inadaptées pour les traiter.

Dans cette section, nous apportons quelques réflexions sur le sujet afin de mettre à profit la pyramide irrégulière.

6.4.1 Définition et classification

La texture joue un rôle très important dans la perception des environnements naturels mais également des objets manufacturés.

La norme *ANSI* définit la texture comme étant une déviation répétitive ou aléatoire par rapport à la surface nominale qui forme la topographie 3D [ANS85].

Le *Petit Larousse* propose plusieurs définitions de la texture en fonction du domaine étudié. Nous retenons celle qui s'applique à la géologie : la **texture** est l'ensemble des caractères définissant l'agencement et les relations volumiques et spatiales des minéraux d'une roche.

Dans le cas de la texture dans les images, l'**ensemble de caractères** est représenté par les attributs des composantes de la texture, l'**agencement** par la structure globale et la **relation** par les configurations locales.

Citons deux classifications des textures afin de mettre l'accent sur leurs caractéristiques fondamentales. Les textures peuvent donc être classées :

- Selon leur échelle : microtexture (grain trop fin pour permettre la perception d'entités) ou macrotexture (émergence d'entités de taille perceptible).
- Selon leur répétabilité : lorsqu'elle est aléatoire, on parlera de textures désordonnées, caractérisées par leur rugosité. Une forte répétabilité correspondra à des textures ordonnées, gérées essentiellement par des relations géométriques et topographiques entre leurs constituants.

Le cadre trop rigide de cette classification quelque peu arbitraire doit prendre en compte une troisième catégorie ‘fourre-tout’ qui concerne les structures faiblement ordonnées : ces dernières également appelées textures orientées sont caractérisées par des orientations locales dominantes qui varient de façon aléatoire.

Classiquement, les textures désordonnées sont étudiées par les approches statistiques alors que les textures ordonnées sont l’objet d’approches structurales. Une étude bien documentée peut être trouvée dans le livre de Rao [Rao90].

6.4.2 Pyramide irrégulière et pavage de Voronoï

Les pyramides irrégulières ont été utilisées très récemment par Lam et Ip [Lam94]. Leur approche combine la pyramide irrégulière et les partitionnements géométriques de Voronoï et de Delaunay. Le pavage de Voronoï a été utilisé pour la première fois en segmentation de texture par Tüceryan [Tuc90]. Lam et Ip restreignent leur méthode aux textures qui sont composées d’éléments de bases assez réguliers (*tokens*) répartis de façon assez régulière sur un fond uniforme : peau de reptile, mur de briques, image de synthèse.

Voici les étapes principales de leur traitement :

1. Binarisation de l’image à traiter pour récupérer les *tokens* et le fond.
2. Séparation des *tokens* supposés connectés.
3. Construction du diagramme de Voronoï généralisé des *tokens* de l’image binarisée.
4. Calcul pour chaque *token* de 5 paramètres relatifs à ses caractéristiques morphologiques : surface, périmètre, facteur de forme, orientation et rapports des deux axes.
5. Calcul de la triangulation de Delaunay duale, qui représente le graphe d’adjacence des *tokens*. Chaque sommet du graphe est alors un *token* muni de ses caractéristiques morphologiques.
6. Construction de la pyramide irrégulière sur le graphe de Delaunay obtenu, en utilisant des critères de fusion évalués sur la similarité des caractéristiques morphologiques des *tokens*.

Remarquons l’adaptabilité de la pyramide irrégulière en fonction du ‘système’ à modéliser :

- les sommets du graphe de la base de pyramide ne sont pas des pixels mais des régions ;
- les attributs utilisés pour caractériser chaque sommet sont d’ordre morphologique.

6.4.3 Pyramide irrégulière et topologie

Le *Petit Larousse* définit la **tesselle** comme un petit morceau de marbre, de pierre, de pâte de verre, de céramique, etc., élément de base d'une mosaïque murale ou d'un pavement.

La tesselle, par définition homogène, constitue l'élément de base de notre étude des textures. Une tesselle peut être caractérisée par de nombreux attributs. Parmi ceux-ci, nous retiendrons :

- sa surface ;
- son facteur de forme ;
- son niveau de gris.

Une classification des tesselles en fonction de la valeur de leurs attributs permet d'associer un type à chaque tesselle. Située dans son environnement, chaque tesselle est caractérisée également par :

- le nombre de tesselles voisines ;
- le type des tesselles voisines ;

Globalement, une texture est caractérisée par le nombre de type des tesselles qui la composent ainsi que par l'arrangement de ces types.

Dans notre approche, nous supposons que chaque tesselle possède une homogénéité importante ainsi qu'une dynamique interne peu élevée. Dans les images de texture, la population de tesselles peut être obtenue par la construction d'une pyramide irrégulière : la croissance en parallèle de région permet de faire émerger en quelques niveaux des régions qui sont autant de tesselles. Un seuil global de similarité adapté à l'image (de valeur assez faible) permet de conserver l'intégrité de chaque tesselles. La segmentation obtenue à l'apex de la pyramide correspond au graphe d'adjacence de la population de tesselles (figure 86) grâce auquel tous les attributs nécessaires à l'étude des textures pourront être étudiés.

Par la suite, une modélisation de la texture ainsi que l'utilisation hiérarchique des attributs des tesselles permettront de reconstituer la composante connexe d'une ou plusieurs textures.

6.5 Discussion et conclusion

Nous avons introduit ce chapitre en montrant que les notions de pyramide et de hiérarchie ne sont pas synonymes de multirésolution.

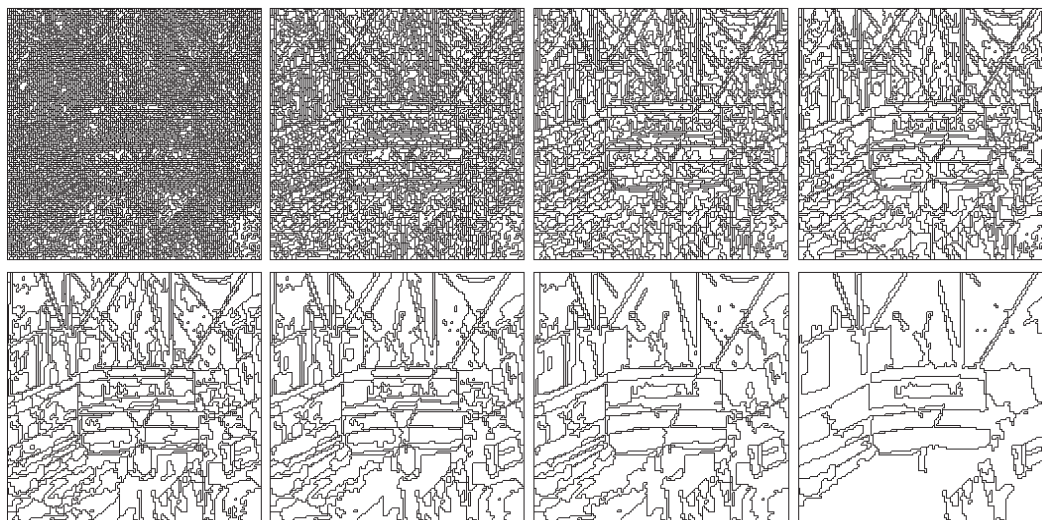
En revanche, la structure pyramidale irrégulière dispose d'éléments très intéressants pour la mise en place de véritables approches multirésolution.

Nous avons ouvert une première voie en utilisant une approche classique de traitement *top-down* récursif, appliqué itérativement à l'aide d'un critère qui joue le rôle

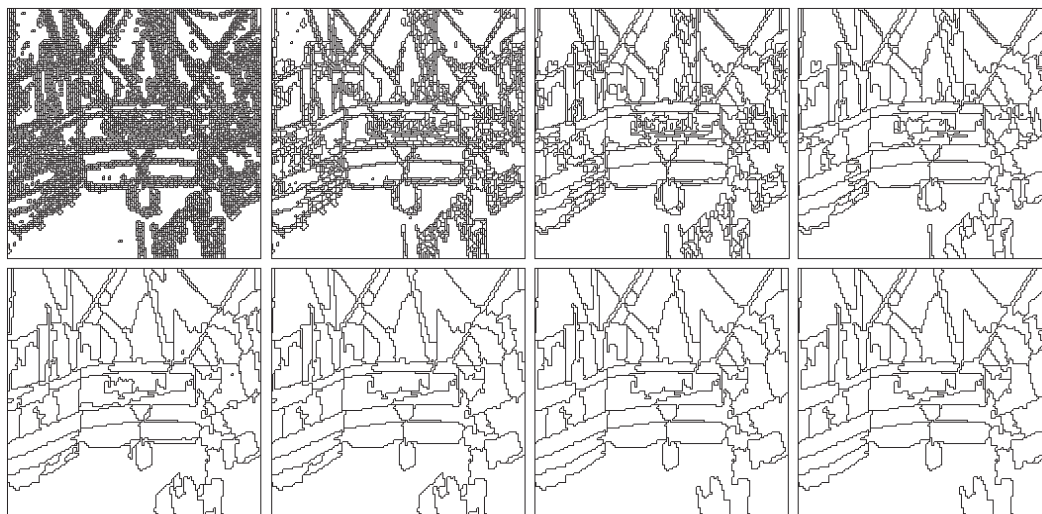
d'un paramètre d'échelle. Cette méthode permet d'obtenir des résultats pouvant être qualifiés de segmentation multirésolution. Les traitements effectués en temps réel permettent à l'utilisateur de parcourir les différentes résolutions et de sélectionner les entités de l'image lorsque leur représentation est optimale.

Ensuite, nous avons développé une technique fondée sur la décimation d'ordre étendu qui permet de localiser des germes dans l'image et de les faire croître. Cette méthode permet de construire une réelle pyramide d'images multirésolution et donne des résultats tout à fait encourageants. Afin de canaliser la croissance de germes souvent désordonnée et qui génère des frontières mal localisées, nous envisageons d'utiliser la coopération région-contour.

Enfin, nous donnons quelques éléments sur les apports de la pyramide irrégulière au problème des textures.



(a) *Pyramide traditionnelle sans multirésolution. La pyramide tronquée comporte 12 niveaux. Ici sont représentés les 7 premiers niveaux et le dernier, avec respectivement 7806, 3605, 1803, 1013, 613, 363, 255 et 94 régions*



(b) *Pyramide multirésolution. La pyramide entière est représentée. Les 8 niveaux comptent respectivement 5751, 1802, 531, 195, 103, 75, 64 et 63 régions*

Fig. 84 - Pyramide classique vs pyramide multirésolution

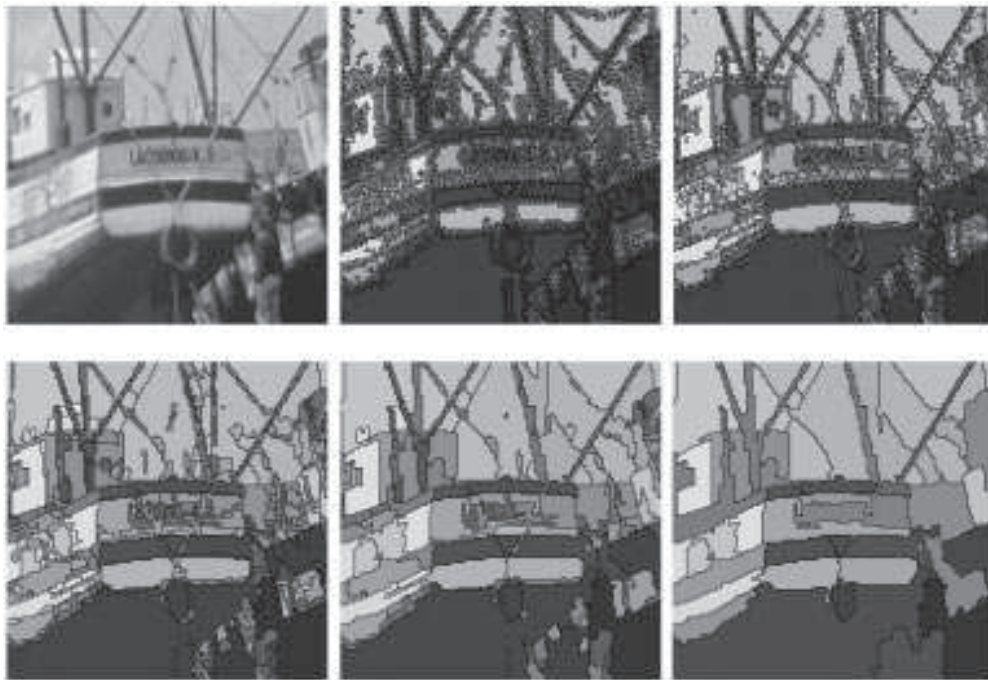
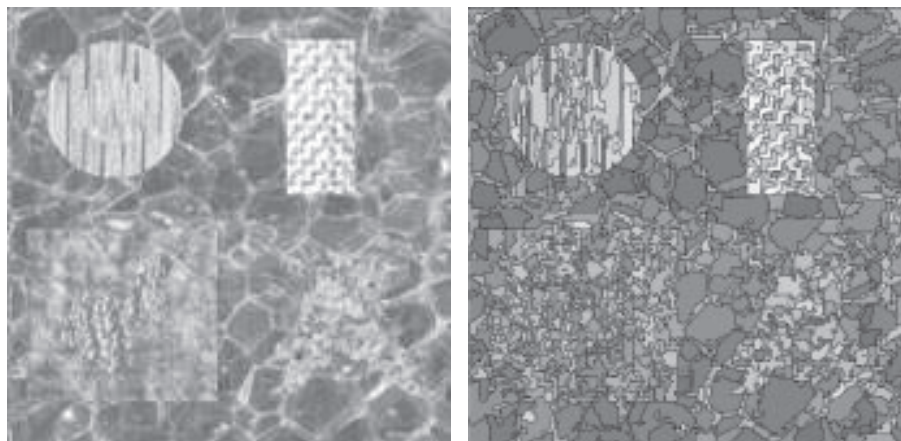


Fig. 85 - Pyramide multirésolution obtenue par décimation étendue



(a) Image de textures

(b) Segmentation des tesselles

Fig. 86 - Extraction des tesselles de plusieurs textures

Chapitre 7

Résultats comparatifs

Pour présenter nos résultats expérimentaux, nous avons choisi six images tests de la banque du GDR TDSI. Ces images regroupent quatre domaines d'activité (bio-médical, satellitaire, objets manufacturés et scène d'intérieur) et comportent chacune une ou plusieurs caractéristiques qui rendent difficiles le traitement d'image en général et la segmentation en particulier.

Un ouvrage très détaillé fournit les résultats obtenus avec les principales méthodes de régions et contours pour ces mêmes images [Coc95a]. Une synthèse succincte sur le même sujet est présentée dans [Coc95b].

7.1 Critères d'évaluation

Afin de déterminer la qualité des résultats d'une segmentation, des critères sont nécessaires. Ceux-ci peuvent être objectifs, fondés sur des calculs statistiques, ou subjectifs, laissés à l'appréciation de l'expertise humaine.

Nous avons choisi de comparer qualitativement les résultats obtenus. La qualité d'une segmentation est relative au domaine d'application et donc à l'usage qui est fait des résultats. Pour un type d'image donné, il est vrai que la quantification du résultat (à travers le calcul d'une erreur quadratique ou d'une distance à la segmentation optimale [Fal93]) peut être significative.

Néanmoins, notre système visuel permet de faire ressortir des caractéristiques non quantifiables qui sont tout aussi importantes pour juger les points forts et les points faibles d'une méthode donnée appliquée à une image particulière.

7.2 Description des images

Voici quelques commentaires qualitatifs et subjectifs que nous avons retenus, aux vues de l'observation des images, de leur histogramme des niveaux de gris (figure 87) et de leur profils (figure 88) :

1. L'image d'**angiographie** (figure 89.a) a été obtenue à partir d'une radiographie analogique d'un réseau vasculaire. Le détachement des veines et des artères sur les autres organes est obtenue grâce à l'injection d'un produit de contraste (opaque aux rayons X) dans les veines du patient.

Le flou de l'image est dû au procédé analogique de radiographie. Le fond est de teinte assez sombre et bruité. Les vaisseaux varient du gris assez sombre (valeur numérique de 75) jusqu'à la saturation dans le blanc. Le problème est essentiellement de conserver la connexion des vaisseaux dont le contraste avec le fond varie le long de leur trajet. Il est d'ailleurs difficile de reconstruire visuellement le réseau. Notons que l'image est également sujet à une altération due vraisemblablement au procédé de développement ou à la manipulation du cliché. Cette image mériterait une segmentation de haut niveau permettant d'effectuer une poursuite des structures filaires.

2. L'image aérienne **Aquitaine** de la campagne du bassin Aquitain (figure 90.a) est une image infrarouge traitée telle-quelle mais rehaussée pour la visualisation : en effet, l'étendue des niveaux de gris est très faible, environ une trentaine d'unités dans les teintes sombres. De ce fait, l'image est en réalité peu bruitée et la plupart des discontinuités sont significatives. On peut voir que la connexité dépend bien de la résolution, si on considère les nombreuses régions filiformes qui correspondent vraisemblablement à des haies, des chemins et des cours d'eau. L'image est altérée par une trame horizontale due sans doute au

capteur utilisé. Cette image pose le problème de la multirésolution : la quantité de détails peut avoir de l'importance en fonction du type d'information à extraire.

3. L'image **bagues** (figure 91.a) semble assez bien contrastée. Néanmoins, elle est très saturée dans le noir avec un phénomène de texture sur une partie du fond. On peut noter également les forts dégradés sur les bagues ainsi que les changements très rapides de type de contours sur la partie extérieure des bagues, causés par l'éclairage et les reflets.
4. L'image du **couloir** (figure 92.a) présente un histogramme particulier qui montre que certains niveaux de gris sont privilégiés par rapport à leurs voisins. L'image sature dans le noir et dans le blanc. Les conditions d'éclairage font que l'image est constituée essentiellement de dégradés qui mettent à mal, en général, les segmentations par approche 'régions'. C'est pourquoi cette image est intéressante.
5. L'image de résonnance magnétique **IRM** montre la coupe d'un torse sur un fond (figure 93.a). Elle est très bruitée, sature dans le noir, et les différents organes sont difficilement segmentables, même visuellement.
6. L'image du **muscle** (figure 94.a) a été obtenue avec un microscope. Elle représente des fibres musculaires claires et sombres sur un fond très clair. Les fibres claires sont très texturées à l'inverse des fibres sombres. La séparation entre deux fibres voisines de même type semble difficile à effectuer de façon automatique : les gradients entre fibres sombres sont assez nets mais de valeur faible, alors que les fibres claires possèdent des frontières souvent inexistantes numériquement. On peut noter des dégradés noyés dans du bruit pour certaines fibres claires, dont la périphérie devient sombre.

Afin d'étayer nos remarques, la figure 87 regroupe l'histogramme de chacune de ces images. La figure 88, quant à elle, montre un profil caractéristique pour chacune des images. Chaque profil correspond aux 256 pixels traversés par la diagonale principale de l'image.

7.3 Résultats et commentaires

Dans cette section, pour plus de clarté, la pyramide irrégulière stochastique est appelée **pyramide classique** ; la méthode de coopération avec le détecteur stochastique **pyramide stochastique**, la méthode de coopération avec l'opérateur de Deriche **coopération Deriche** ; la méthode de croissance de germes **croissance de germes**.

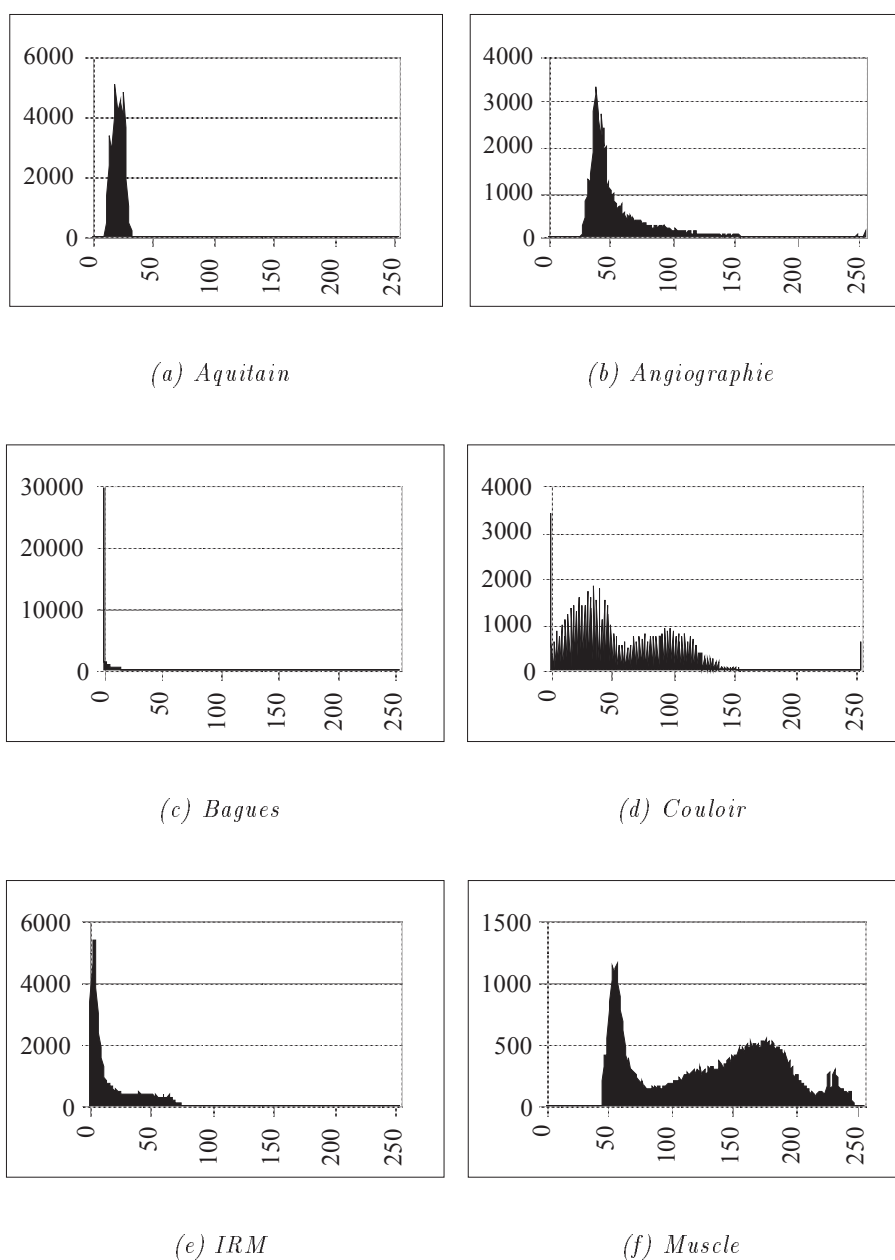


Fig. 87 - Histogramme des niveaux de gris des images utilisées pour les tests

Toutes les segmentations possèdent moins de 250 régions. Parmi les différents résultats obtenus lors de nos essais, nous avons retenu ceux qui donnaient le meilleur rapport qualité/nombre de régions.

Aquitaine Comme les régions sont relativement petites et homogènes, la pyramide

irrégulière classique réussit assez bien. Des résultats plus précis peuvent être obtenus avec un seuil de similarité T_g plus faible. La coopération stochastique offre un résultat avec peu de régions. Quelques structures très fines sont très bien détectées. La coopération Deriche donne un résultat globalement satisfaisant avec une très bonne fidélité des petites régions contrastées. La croissance de germes génère quelques régions avec des frontières mal localisées.

Angiographie La pyramide irrégulière est gênée par le bruit, et le dégradé flou des vaisseaux. Peu de connexions sont conservées. La coopération stochastique préserve mieux les connexions, et est moins sensible au bruit et aux dégradés. La coopération Deriche est la seule à fournir des entités assez bien connectées et filiformes. Les régions sont peu nombreuses. La croissance de germes qui donne peu de régions, est peu sensible au bruit. Les connexions sont assez bien conservées mais en plusieurs régions connexes.

Bagues est une image où les résultats diffèrent assez peu. La pyramide classique est celle qui déconnecte le plus les bagues. La coopération stochastique déconnecte un peu mais isole beaucoup de petites régions fines et rectilignes ainsi que plusieurs bagues. Malgré la détection de nombreux reflets, la coopération Deriche préserve un maximum de connexions. La croissance de germes quant à elle, propose plusieurs bagues distinctes mais elle est sensible aux reflets.

Couloir La pyramide classique fournit peu de structures rectilignes car elle est très gênée par les dégradés qui forment des frontières arbitraires. La coopération stochastique est très peu gênée par les dégradés. Elle conserve un grand nombre de structures rectilignes et de détails bien localisés. La coopération Deriche et la croissance de germes sont un peu gênée par les dégradés. La première fournit des structures qui ne sont pas très nettes, alors qu'un nombre important de régions sont arbitrairement découpées sur le plafond avec la deuxième.

IRM Avec la pyramide classique, le fond qui contient du bruit est sur-segmenté alors que le torse qui contient l'information est sous-segmenté. La coopération stochastique est peu sensible au bruit et permet de distinguer différents organes. La coopération Deriche propose un résultat net, sans bruit de fond et avec quelques organes bien segmentés. La croissance de germes est très gênée par le bruit sur le torse. De nombreux découpages sont arbitraires.

Muscle La pyramide classique n'a gardé que très peu d'information dans les fibres claires. Les fibres sombres ont pour la plupart fusionné. La coopération stochastique arrive à segmenter assez bien les fibres sombres, certaines fibres claires et les petites zones de fond sont assez fidèles. La coopération Deriche conserve l'information de façon assez constante sur toute l'image. On peut noter l'artéfact causé par les contours qui ne joignent pas le bord de l'image. La croissance de germes fournit quelques fibres mais les frontières sont mal localisées.

En conclusion, la pyramide classique fournit de bons résultats sur des images sans dégradés et peu bruitées. Les méthodes coopératives, quant à elles, sont peu sensibles

au bruit et aux dégradés. En effet, la coopération Deriche donne les résultats les plus nets alors que la coopération stochastique est capable de fournir plus de détails. En ce qui concerne la croissance de germes, elle permet d'obtenir des segmentations avec un nombre prédéfini de régions mais reste très sensible au bruit. Néanmoins, cette dernière approche est essentiellement orientée vers la multirésolution, ce qui la distingue de la finalité des autres méthodes.

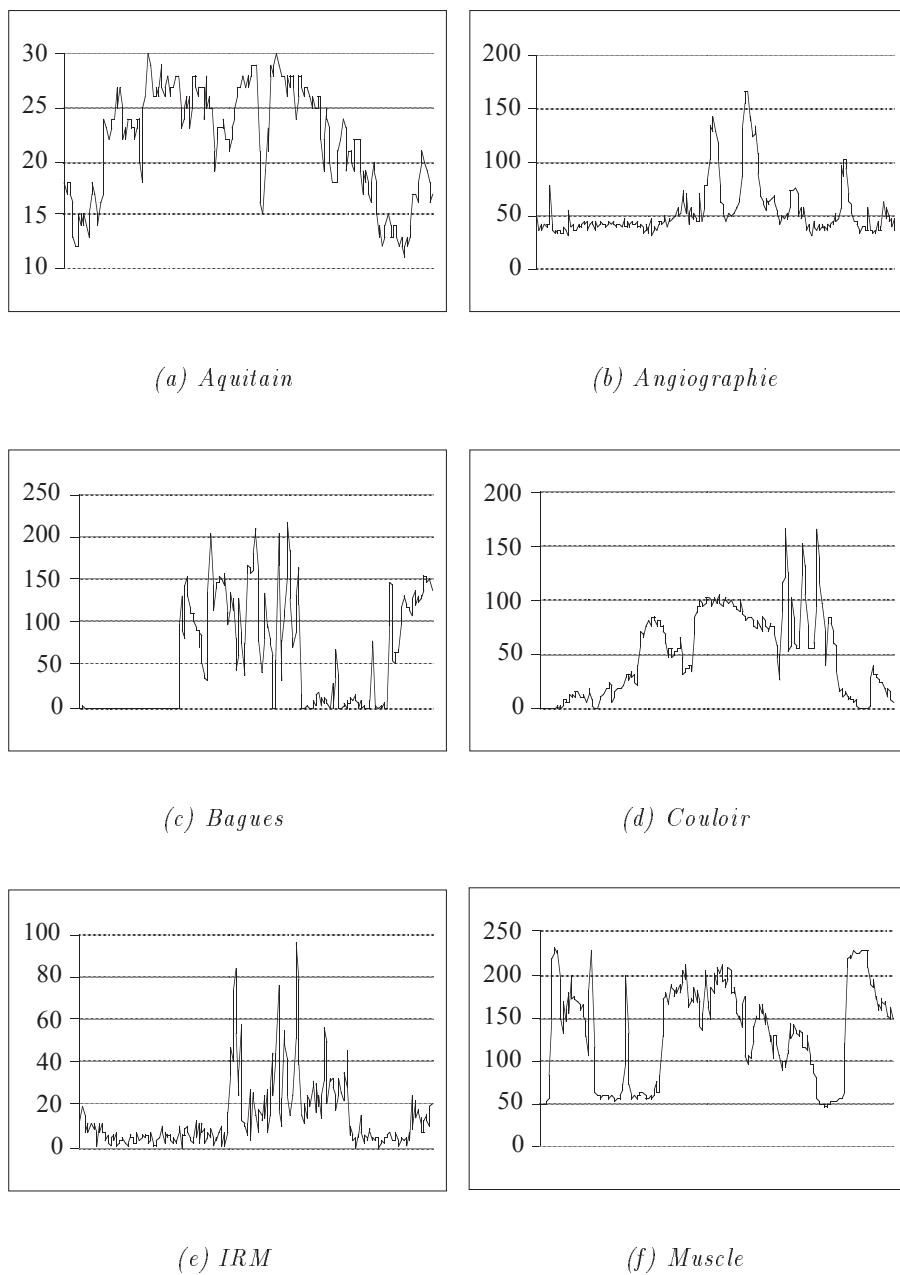
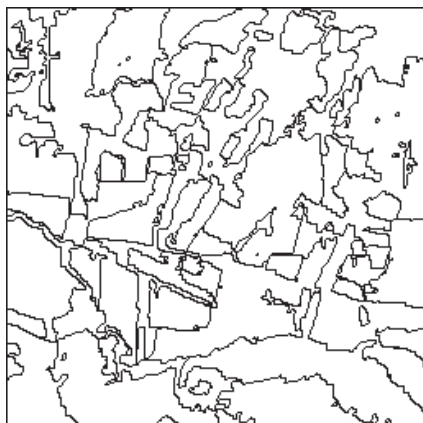


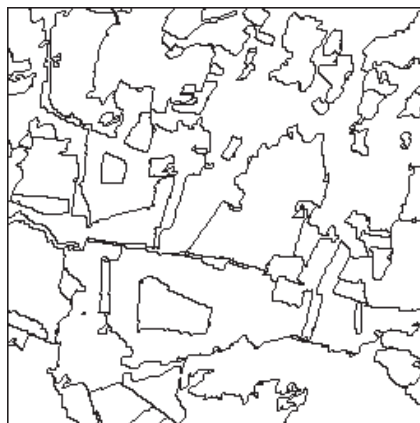
Fig. 88 - Profils significatifs des images utilisées pour les tests (profils des diagonales principales)



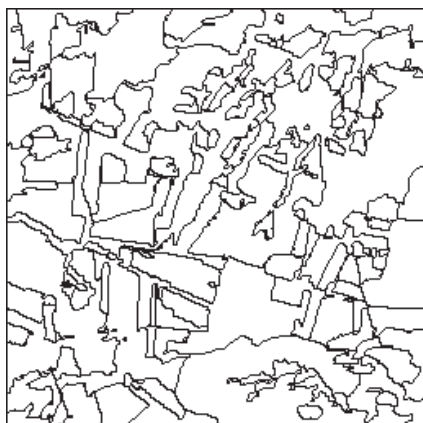
(a) Image originale (rehaussée)



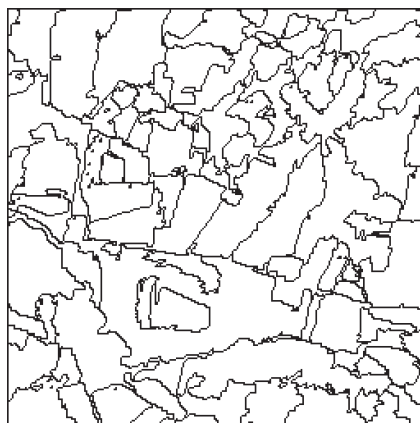
(b) Pyramide stochastique



(c) Coopération 'stochastique'

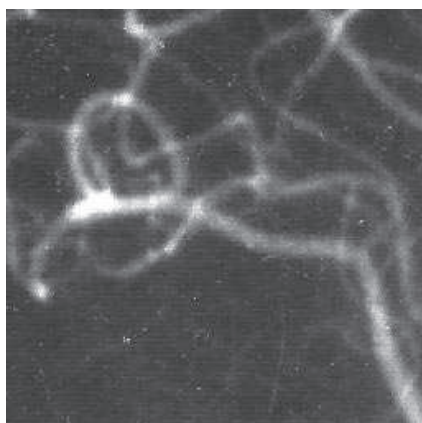


(d) Coopération 'Deriche'

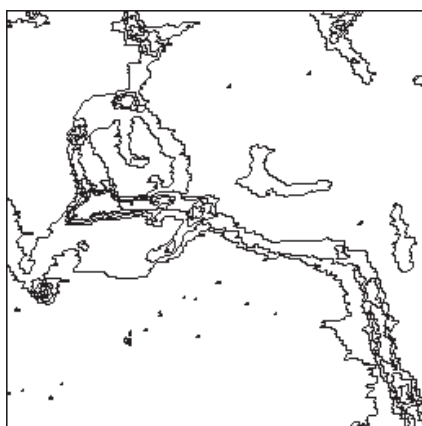


(e) Croissance de germes

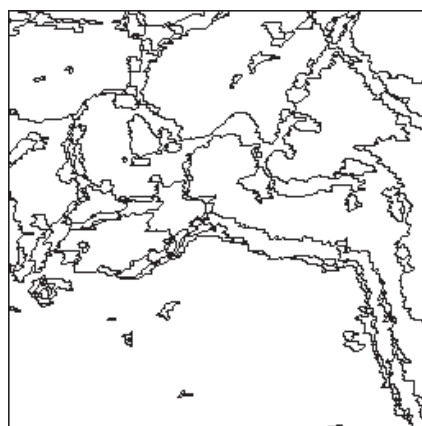
Fig. 89 - Résultats de segmentation pour l'image 'Aquitaine'



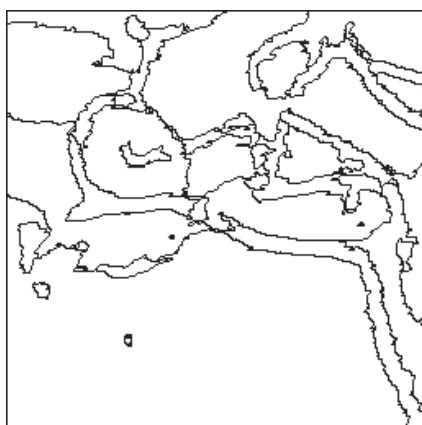
(a) Image originale



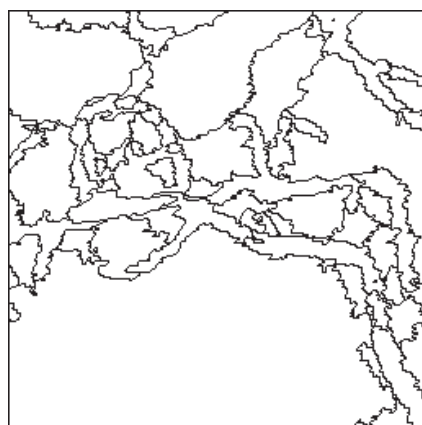
(b) Pyramide stochastique



(c) Coopération 'stochastique'

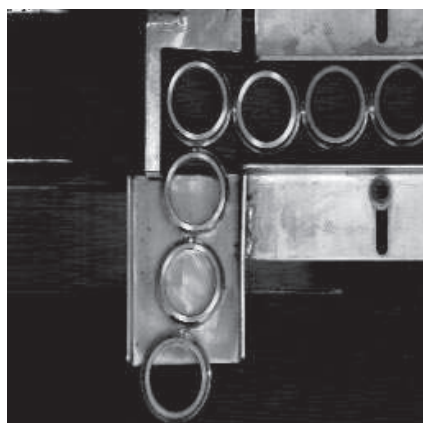


(d) Coopération 'Deriche'

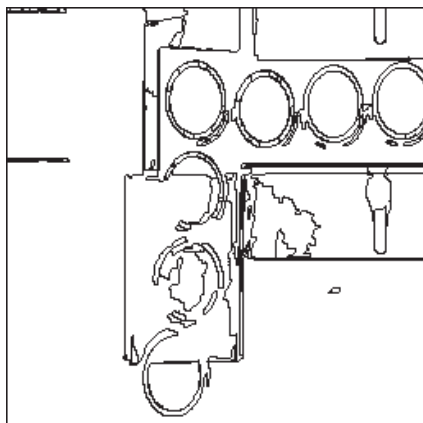


(e) Croissance de germes

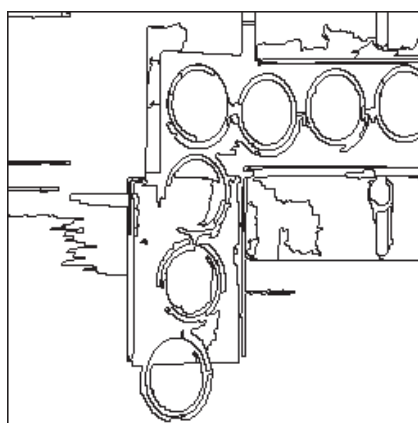
Fig. 90 - Résultats de segmentation pour l'image 'Angiographie'



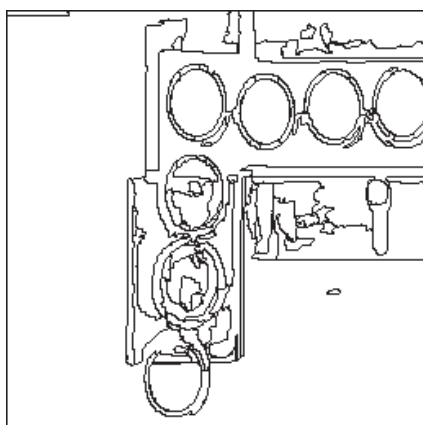
(a) Image originale



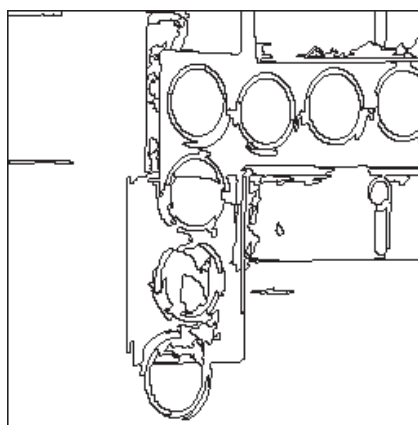
(b) Pyramide stochastique



(c) Coopération 'stochastique'



(d) Coopération 'Deriche'

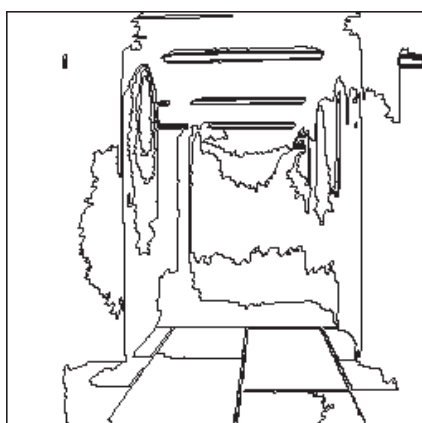


(e) Croissance de germes

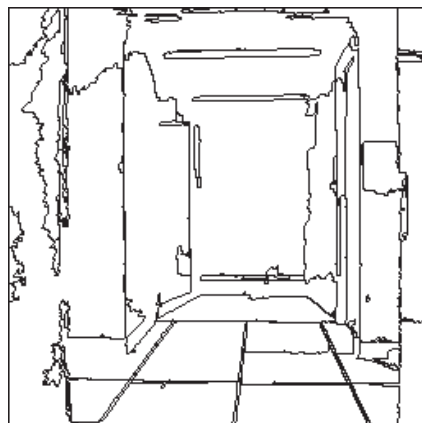
Fig. 91 - Résultats de segmentation pour l'image 'Bagues'



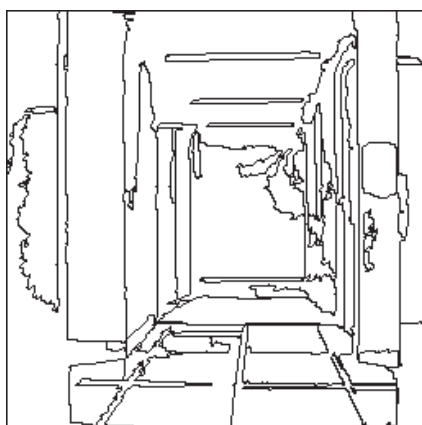
(a) Image originale



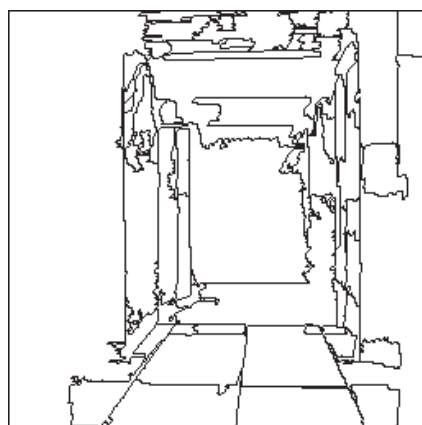
(b) Pyramide stochastique



(c) Coopération 'stochastique'

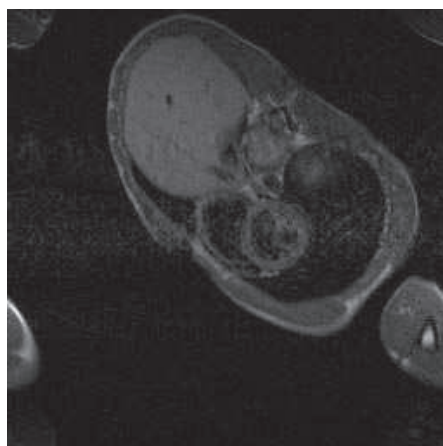


(d) Coopération 'Deriche'

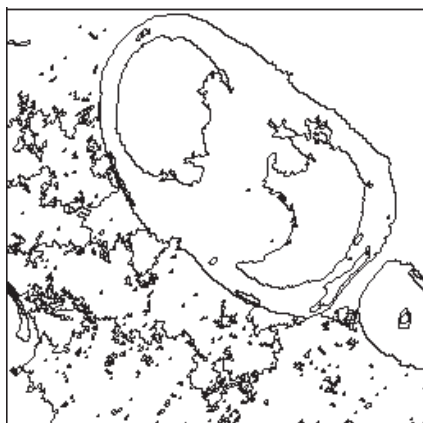


(e) Croissance de germes

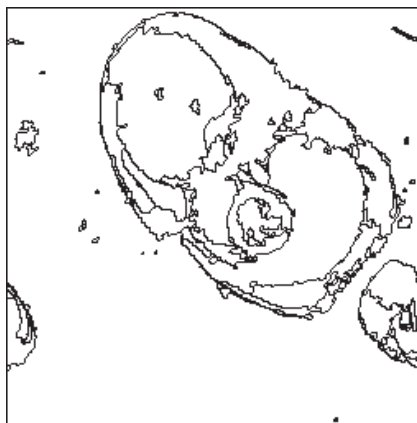
Fig. 92 - Résultats de segmentation pour l'image 'Couloir'



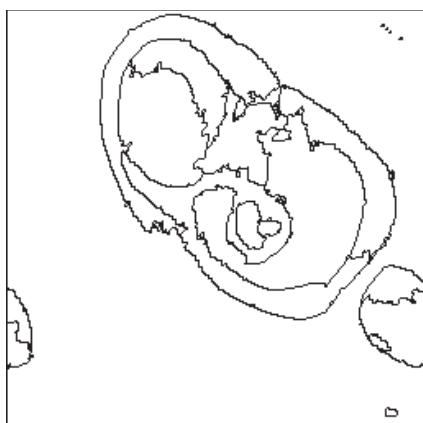
(a) Image originale



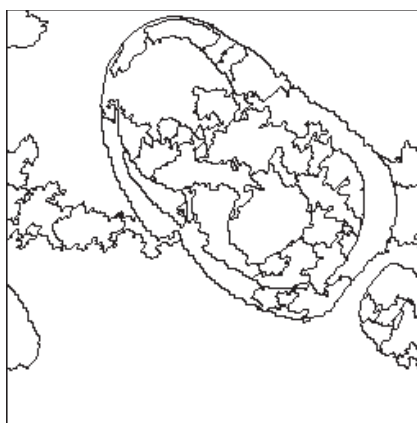
(b) Pyramide stochastique



(c) Coopération 'stochastique'

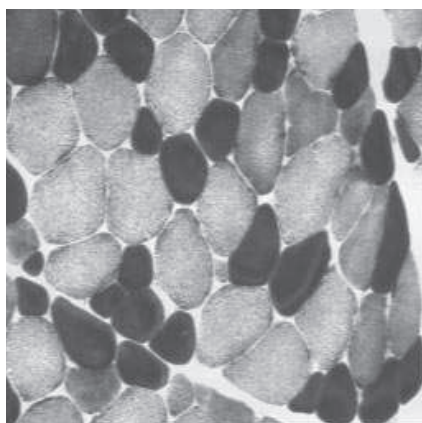


(d) Coopération 'Deriche'

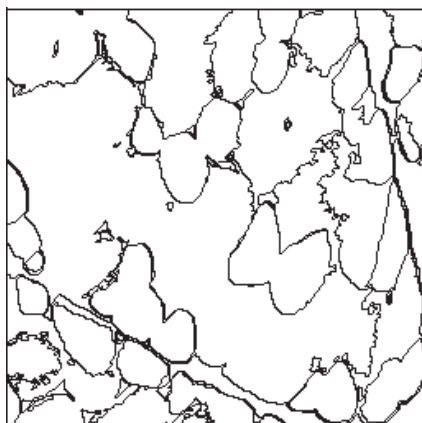


(e) Croissance de germes

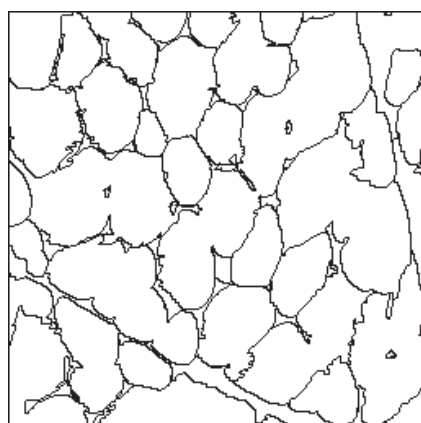
Fig. 93 - Résultats de segmentation pour l'image 'IRM'



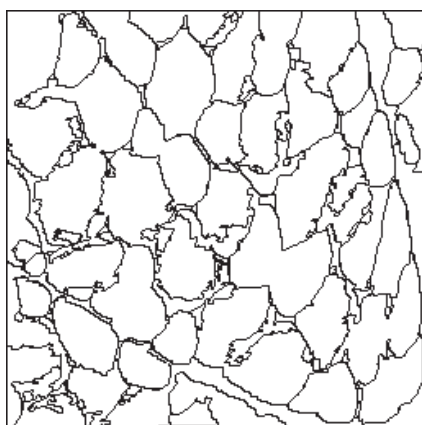
(a) Image originale



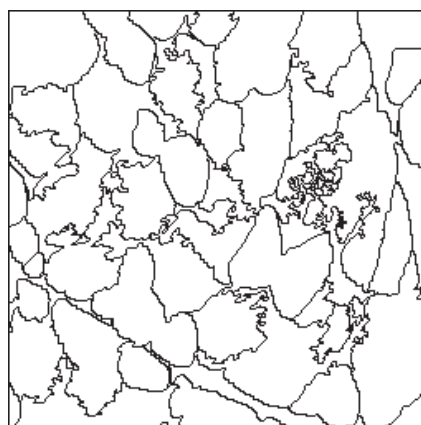
(b) Pyramide stochastique



(c) Coopération 'stochastique'



(d) Coopération 'Deriche'



(e) Croissance de germes

Fig. 94 - Résultats de segmentation pour l'image 'Muscle'

Chapitre 8

Conclusion



Fig. 95 - Pyramides de Guizèh

8.1 Bilan

Nous avons présenté (chapitre 2) un état de l'art des modèles de multirésolution dans différents domaines du traitement d'image et notamment en segmentation.

Par la suite (chapitre 3), nous avons fait un rapide aperçu de différentes techniques de segmentation.

Munis d'une vue d'ensemble des outils dont dispose la segmentation d'image, nous avons décrit (chapitre 4) les différentes phases de construction de la pyramide irrégulière pour segmenter les images. Cette description est complétée par une étude expérimentale des caractéristiques de la méthode.

En partant d'un constat sur la qualité des segmentations obtenues et sur les améliorations souhaitées, nous avons apporté notre contribution sur les points suivants :

- Nous avons mis en place une méthode de coopération régions-contours (chapitre 5) qui respecte la structure de graphes. Cette coopération peut être effectuée avec la majorité des opérateurs de contours existants.
- Pour cette coopération de processus, nous avons imaginé un détecteur de contours stochastique adaptatif pour tenter de servir au mieux la croissance en parallèle de régions.
- Dans le but de rechercher une meilleure exploitation de la multirésolution dans la pyramide irrégulière, nous avons développé (§ 6.2) une première méthode. Celle-ci a permis, à l'aide des différents niveaux d'une pyramide irrégulière classique, de fournir rapidement un nombre important de segmentations dérivées, chacune à une résolution différente.
- Par la suite (§ 6.3), en utilisant la notion de décimation étendue, nous avons construit directement une pyramide d'images à différentes résolutions de détails.
- Quelques éléments de base ont été proposés pour l'étude des images texturées. Celles-ci pourraient grandement tirer partie de la structure de graphe et de la croissance en parallèle de régions offertes par la pyramide irrégulière.
- Enfin, nous avons fait un comparatif qualitatif des résultats obtenus avec la pyramide originale et les méthodes proposées (chapitre 7).

Notre bilan est très positif, il peut se résumer par les points suivants :

- La bonne adéquation de la structure pyramidale avec les contours ainsi que les résultats obtenus nous permettent d'affirmer que le principe d'intégration est capital en segmentation d'image.
- Nous avons élargi une voie dans le domaine de la segmentation multirésolution qui, à l'heure actuelle, est peu étudié. Malgré tout, celui-ci constitue sans doute une des clés de la réussite de la compréhension des images.

- La diversité des méthodes développées montrent une fois de plus le potentiel très élevé de la structure pyramidale irrégulière qui repose néanmoins sur un formalisme simple.

8.2 Perspectives

Notre travail personnel a été guidé par deux fils conducteurs :

- améliorer les résultats obtenus avec la structure pyramidale irrégulière existante, par la mise en place d'une approche coopérative ;
- proposer une nouvelle approche de la multirésolution par le biais de la segmentation.

La pyramide irrégulière permet d'envisager un grand éventail de recherches. Les problématiques qui nous paraissent d'un intérêt majeur et qui demeurent largement ouvertes sont les suivantes :

- L'optimisation de la construction de la pyramide irrégulière afin de proposer aux utilisateurs des segmentations multirésolutions temps-réels qui pourraient dans de nombreux cas remplacer le traditionnel seuillage binaire.
- Comme étudiée au chapitre 6, la pyramide irrégulière est un outil puissant et souple qui permettrait une meilleure prise en compte de l'aspect topologique des images de texture.
- Le traitement des séquences d'images est très intéressant. Différents types d'approches pyramidales régulières sont déjà proposés dans la littérature.

Après la construction d'une pyramide irrégulière sur la première image d'une séquence, l'utilisation des multiples partitionnements obtenus offrirait un modèle souple et adaptatif pour les images suivantes. Le passage d'une image à la suivante effectué à des résolutions plus ou moins faibles permettrait de réaliser du suivi d'objets sans avoir à reconstruire la pyramide depuis sa base.

- L'implantation sur machines parallèles poserait de gros problèmes liés à l'aspect irrégulier de la structure. Néanmoins, ce travail montrerait la réelle supériorité des traitements jusqu'alors utilisés sur des architectures séquentielles.

Bibliographie

- [Ada94] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, june 1994.
- [ANS85] Surface texture surface roughness, waviness, and lay. *The American National Standards Institute*, 46(1), june 1985.
- [Att91] G. Attolico, A. Distanto, T. d’Orazio, and E. Stella. An integration of edge and region based methods for the segmentation of needle maps. In V. Cantoni and al, editors, *Progress in Image Analysis and Processing II*, pages 302–309. 6th ICIAP, Como, Italy, 1991.
- [Bau92] O. Baujard. *Conception d’un environnement de développement pour la résolution de problèmes : Apport de l’Intelligence Artificielle Distribuée à la vision*. PhD thesis, Université Joseph Fourier, Grenoble, 1992.
- [Ber92] P. Bertolino and A. Montanvert. Edge detection for biomedical image: a self-adaptive and randomized operator. In *Proc. of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE Comp. Soc. Press*, pages 129–133, Paris, November 1992.
- [Ber94] E. Bertin. *Diagrammes de Voronoï 2D et 3D : Applications en analyse d’images*. PhD thesis, Université Joseph Fourier, Grenoble, 1994.
- [Ber95a] E. Bertin, H. Bischof, and P. Bertolino. Voronoi pyramid controlled by Hopfield networks. *Computer Vision Graphics and Image Processing*, 1995. A paraître.
- [Ber95b] P. Bertolino and A. Montanvert. Stochastic edge detection based on discrete segments. In *Actes du 5ème Congrès DGCI*, pages 117–126, Clermont-Ferrand, 25-27 Septembre 1995.
- [Ber96] P. Bertolino and A. Montanvert. Coopération régions-contours multirésolution en segmentation d’image. In *Actes du 10ème Congrès RFIA*, pages 299–307, Rennes, 16-18 Janvier 1996. Accepted.
- [Bev89] J.R. Beveridge, J. Griffith, R. Kohler, A. Hanson, and E. Riseman. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision*, 29(2):311–347, 1989.

- [Bis90] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617, September 1990.
- [Bis94] H. Bischof, E. Bertin, and P. Bertolino. Voronoi pyramid and hopfield networks. In *proc of the 12th ICPR*, Jerusalem, Israel, 9-13 October 1994.
- [Bla87] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge Mass., 1987.
- [Bla88] R. P. Blanford and S. L. Tanimoto. Bright-spot detection in pyramids. *Computer Vision Graphics and Image Processing*, 43:133–149, 1988.
- [Bow81] A. Bowyer. Computing dirichlet tessellation. *The Computer Journal*, 24:162–166, 1981.
- [Bra95] G. Braviano. *Logique floue en analyse d'image*. PhD thesis, Université Joseph Fourier, Grenoble, 1995.
- [Bre65] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM System Journal*, 4:25–30, 1965.
- [Buc88] C.E. Buckley. Split-and-merge image segmentation based on Delaunay triangulation. In *Lectures notes in computer science*, 333, *International Workshop Computational Geometry*, Warzburg, March 1988.
- [Bur81a] P.J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51, 1981.
- [Bur81b] P.J. Burt, T.H. Hong, and A. Rosenfeld. Image segmentation and region property computation by cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11:802–809, 1981.
- [Bur83] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31:532–540, 1983.
- [Bur84] P.J. Burt. The pyramid as a structure for efficient computation, multiresolution image processing and analysis. In A. Rosenfeld, editor, *Multiresolution image processing and analysis*, pages 6–35, Berlin, 1984. Springer Verlag.
- [Can83] J.F. Canny. Finding edges and lines in image. Technical Report 720, MIT, Artif. Intel. Lab., Cambridge, 1983.
- [Can85] V. Cantoni, M. Ferretti, S. Levialdi, and F. Maloberti. A pyramidal project using integrated technology. In S. Levialdi, editor, *Integrated Technology for Parallel Image Processing*, Londres, Academic Press, 1985.

- [Can86] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [Cas89] S. Castan, J. Zhao, and J. Shen. Une famille de détecteurs de contours basée sur le filtre exponentiel optimal. In *AFCET RFIA89*, Paris, 1989.
- [Cha85] J.-M. Chassery. Deux algorithmes orientés parallélisme : courbes de niveau et étiquetage. In *AFCET-INRIA*, editor, *Proc. du 5ème Congrès RFIA*, pages 541–548, Grenoble, Novembre 1985.
- [Cha89] F. Chantemargue, M. Popovic, R. Canals, and P. Bonton. Parallelization of the merging step of the region segmentation method. In *7th Scandinavian Conference on Image Analysis*, pages 993–940, Aalborg, Denmark, August 13-16 1989.
- [Cha91] J.-M. Chassery and A. Montanvert. *Géométrie Discrète en Analyse d'Images*. Edition Hermès, Paris, 1991.
- [Ché92] A. Chéhikian. Algorithmes optimaux pour la génération de pyramides d'images passe-bas et laplaciennes. *Traitement du Signal*, 9(4):297–307, 1992.
- [Chi94] E. Chiarello. *Pyramide stochastique et écologie du paysage*. PhD thesis, Université Claude Bernard, Lyon, 1994.
- [Cho95] K. Cho and P. Meer. Image segmentation from consensus information. Technical Report CE-124, Department of Electrical and Computer Engineering, Rutgers University, New Jersey, USA, September 1995.
- [Cib84] J. Cibulskis and C.R. Dyer. Node linking strategies in pyramids for image segmentation. In A. Rosenfeld, editor, *Multiresolution image processing and analysis*, pages 109–120, Berlin, 1984. Springer Verlag.
- [Cle91] P. Clermont and S. Cartier. Implementation and evaluation of region growing algorithms on cm2. In *Proc. of the Int. Colloquium on Parallel Image Processing*, pages 91–109, Paris, France, June 17-19 1991.
- [Coc95a] J-P. Cocquerez, S. Philipp, P. Bolon, J-M. Chassery, D. Demigny, C. Gracfigne, A. Montanvert, R. Zéboudj, and J. Zérubia. *Analyse d'images : filtrage et segmentation*. Paris, 1995.
- [Coc95b] J-P. Cocquerez, S. Philipp, and R. Zéboudj. Comparaison de méthodes de segmentation d'images. In B. Meltzer and D. Michie, editors, *Actes de 15ème colloque GRETSI*, pages 1355–1360, Juans-les-Pins, 18-21 septembre 1995.
- [Cro84] J.L. Crowley and R.M. Stern. Fast computation of the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:212–222, 1984.

- [Den88] J. Dengler and M. Schmidt. The dynamic pyramid. a model for the motion analysis with controlled continuity. In V. Cantoni and al, editors, *Image Analysis and Processing II*, pages 337–344. 4th ICIAP, Palermo, Italy, 1988.
- [Der87] R. Deriche. Using Canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, pages 167–187, 1987.
- [Dor84] L. Dorst and X. Duin. Spirograph theory: a framework for calculations on digitized straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):632–639, September 1984.
- [dW85] G.S. Van der Wal and J.O. Sinniger. Real time pyramid transform architecture. In *SPIE Proc. of Intelligent Robots and Computer Vision*, pages 300–305, Boston, 1985.
- [Dye87] C.R. Dyer. Multiscale image understanding. In Leonard Uhr, editor, *Parallel Computer Vision*, pages 171–213. Academic Press, Inc, Orlando, Florida, 1987.
- [Elo94] Y. Elomary. *Modèles déformables et multi-résolution pour la détection de contours*. PhD thesis, Université Joseph Fourier, Grenoble, 1994.
- [Fal93] R.K. Falah and Ph. Bolon. Mesure de dissimilarité entre deux segmentations. In *14ième colloque GRETSI*, pages 763–766, Juan-les-Pins, Septembre 1993.
- [Fal94] R.K. Falah, Ph. Bolon, and J.P. Cocquerez. A region-region and region-edge cooperative approach of image segmentation. In *Proc. IEEE ICIP conference*, volume 3, pages 470–474, Austin, 1994.
- [Fis58] W.D. Fisher. On grouping for maximum homogeneity. *JASA*, 53:789–798, 1958.
- [Gag85] A. Gagalowicz and O. Monga. Un algorithme de segmentation hiérarchique. In AFCET-INRIA, editor, *Actes du 5ème Congrès RFIA*, pages 163–177, Grenoble, France, Novembre 1985.
- [Gag86] A. Gagalowicz and O. Monga. A new approach for image segmentation. In *8th ICPR a verifier*, pages 265–267, Paris, France, 1986.
- [Gei91] D. Geiger and F. Girosi. Parallel and deterministic algorithms on mrf. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), May 1991.
- [GRE91] GRECO. *Segmentation première partie : prétraitement et approche frontière*. 1991. Rapport du GDR 134 du CNRS.

- [Han84] W. Handler, U. Herzog, F. Hofmann, and H.J. Schneider. Multiprozessoren für breite answendungsgebiete : Erlangen general purpose array, gi/ntg fachtagung architektur und betrieB von rechensystemen. In Springer-Verlag, editor, *Proc. of the 1st European Conference on Computer Vision*, pages 195–208, Berlin, 1984.
- [Har84] P.N. Hartman and S.L. Tanimoto. An hexagonal pyramid data structure for image processing. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2):247–256, July 1984.
- [Har85] R.M. Haralick and L.G. Shapiro. Survey: image segmentation techniques. *Computer Vision Graphics and Image Processing*, 29:100–132, May 1985.
- [Har87] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:523–550, July 1987.
- [Hon84] T.H. Hong and A. Rosenfeld. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):222–229, march 1984.
- [Hor74] J.L. Horowitz and T. Pavlidis. Picture segmentation by a direct split-and-merge procedure. In *Proc. of the 2nd ICPR*, pages 424–433, 1974.
- [Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. In *Proc. IRE*, volume 40, pages 1098–1101, 1952.
- [Jol90] J.M. Jolion. Analyse d’images : le modèle pyramidal. *Traitement du Signal*, 7(1):5–17, 1990.
- [Jol92] J.M. Jolion and A. Montanvert. The adapted pyramid: a framework for 2d image analysis. *Computer Vision Graphics and Image Processing*, 55(3):339–348, May 1992.
- [Jol94a] J.M. Jolion. Analyse multirésolution du contraste dans les images numériques. *Traitement du Signal*, 11(3):245–255, 1994.
- [Jol94b] J.M. Jolion and A. Rosenfeld. *A pyramid framework for early vision*. Kluwer Academic publishers, 1994.
- [Kas88] M. Kass, A. Witkins, and D. Terzopoulos. Snakes: Active contour models. *Computer Vision Graphics and Image Processing*, pages 321–331, 1988.
- [Kel71] M.D. Kelly. Edge detection in pictures by computer using planning. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 6, pages 397–409, Edinburgh, Scotland, 16-20 septembre 1971. Edinburgh Univ. Press.

- [Kir71] R. Kirsch. Computer determination of the constituent structure of biological images. *Computer Biomedical Research*, 4:315–328, 1971.
- [Koe84] Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [Kon93] H. Konik, M. Calonnier, and B. Laget. Adaptive texture image segmentation using local-base pyramids. In *Proc of the 6th European Congress for Stereology*, volume 13, pages 499–504, Prague, 1993.
- [Kon94] H. Konik. *Contribution de l'approche pyramidale à la segmentation d'images texturées*. PhD thesis, Université Jean Monnet, Saint-Etienne, 1994.
- [Kra91] J. Lopez Krahe. Problème d'arithmétique en analyse d'images et géométrie des formes. In *Premier Colloque Géométrie Discrète en Imagerie: Fondements et Applications*, Strasbourg, France, 26-27 Septembre 1991.
- [Kro85] W.G. Kropatsch. A pyramid that grows by powers of 2. *Pattern Recognition Letters*, 6(3):315–322, 1985. A veriifier.
- [Kro87] W.G. Kropatsch. Curve representations in multiple resolutions. *Pattern Recognition Letters*, 6(3):179–184, August 1987.
- [Kro91a] W.G. Kropatsch. Image pyramids and curves: an overview. Technical Report PRIP-TR-2, Dept. for Pattern Recognition and Image Processing, Institute for Automation, Technical University of Vienna, Austria, March 1991.
- [Kro91b] W.G. Kropatsch and A. Montanvert. Irregular versus regular pyramid structures. In U. Eckhardt, A. H ubler, W. Nagel, and G. Werner, editors, *Geometrical problems of image processing*, pages 11–22, Georgenthal, Germany, march 1991. Akademie Verlag, Berlin.
- [Kro95] W.G. Kropatsch. Towards higher decimation ratios. In *Proc. of the CAIP'95*, 1995.
- [Lam94] S.W.C. Lam and H.H.S.Ip. Structural texture segmentation using irregular pyramid. *Pattern Recognition Letters*, 15(9):691–698, July 1994.
- [Let59] J.Y. Lettvin, H.R. Maturana, W.S. McCulloch, and W.H. Pitts. What the frog's eye tells the frog's brain. *Proc of Instit. Radio Eng.*, 47:1940–1959, 1959.
- [Li93] Z.N. Li and D. Zhang. Fast line detection in a hybrid pyramid. *Pattern Recognition Letters*, 1993.
- [Lin90] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, march 1990.

- [Lub86] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal of Computing*, 15(4):1036–1053, November 1986.
- [Mal89] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
- [Man94] A. Manzanera and J.M. Jolion. Pyramide irrégulière. In *Proc. du 9ème Congrès RFIA*, pages 221–228, 1994.
- [Mar89] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):234–254, july 1989.
- [Mat93] Ch. Mathieu. *Segmentation d’images par pyramides souples : Application à l’imagerie médicale multidimensionnelle*. PhD thesis, Institut National des Sciences Appliquées de Lyon, 1993.
- [Mee89] P. Meer and S. Connelly. A fast parallel method for synthesis of random patterns. *Pattern Recognition*, 22:189–204, 1989.
- [Mee90] P. Meer, D. Mintz, A. Montanvert, and A. Rosenfeld. Consensus vision. In *Proceedings of AAAI-90, Workshop on Qualitative Vision*, pages 111–115, Boston, Mass., July 1990.
- [Mel91] M. Melkemi. *Approches Géométriques par Modèles de Voronoï en Segmentation d’Images*. PhD thesis, Université Joseph Fourier, Grenoble, 1991.
- [Mel92] M. Melkemi and J.M. Chassery. Edge-region segmentation process based on generalized voronoi diagram representation. In *Proc. of the 11th IAPR, International Conference on Pattern Recognition*, volume 3, pages 323–326, The Hague, The Netherlands, September 1992.
- [Mér85] A. Mérigot, B. Zavidovique, and F. Devos. Sphinx, a pyramidal approach to parallel image processing. In *Proc. of CAPAIDM workshop*, pages 107–111, Miami, 1985.
- [Mon89] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. Technical Report CS TR 2322, Computer Vision Laboratory, University of Maryland, September 1989.
- [Mon91] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, April 1991.
- [Mon92a] A. Montanvert and P. Bertolino. Irregular pyramids for parallel image segmentation. In *Proc. of the 16th OAGM Meeting*, pages 13–34, Vienna, Austria, May 6-8 1992.

- [Mon92b] A. Montanvert, P. Meer, and P. Bertolino. Optimal hierarchical shape analysis in gray level images. In *NATO advanced workshop "Shape in Picture"*, volume 126, pages 13–34, Driebergen, The Netherlands, September 7–11 1992. Springer Verlag ed.
- [Naz84] A. Nazif and M.D. Levine. Low-level image segmentation: an expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):555–577, september 1984.
- [Oka92] A. Okabe, B. Boots, and K. Sugihara. *Spatial tessellations. Concepts and applications of Voronoi diagrams*. 1992.
- [Pal93] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
- [Par91] R.-H. Park and P. Meer. Edge-preserving artifact-free smoothing with image pyramids. *Pattern Recognition Letters*, 12:467–475, August 1991.
- [Pav90] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–233, March 1990.
- [Pav92] T. Pavlidis. Why progress in machine vision is so slow. *Pattern Recognition Letters*, 13:221–225, April 1992.
- [Pel82] T. Peli and D. Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20:1–21, April 1982.
- [Pel86] S. Peleg and O. Federbusch. Custom made pyramids. In V. Cantoni and S. Levialdi, editors, *Pyramidal Systems for Computer Vision*, volume F25, pages 165–171, Berlin, Heidelberg, 1986. Springer Verlag.
- [Per87] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *Workshop on Computer Vision*, pages 16–22, Miami, Florida, November, December 1987.
- [Pog88] T. Poggio, J. Little, E. Gamble, W. Gillet, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, H. Bulthoff, M. Drumheller, P. Oppenheimer, W. Yang, and A. Hurlbert. The mit vision machine. In *Proc. of Image Understanding Workshop*, pages 177–198, Cambridge, Mass., April 6–8 1988.
- [Pre70] J.M.S. Prewitt. Object enhancement and extraction. In *Picture Processing and Psychopictorics*, pages 75–149. B.S. Likin and A. Rosenfeld, Academic Press,, 1970.
- [Qui91] T. Quiguer, P. Miché, and R. Debrie. Segmentation by auto-adaptative thresholding. In *Proc. of the 6th International Conference on Image Analysis and Processing*, Como, Italy, 4–6 September 1991.

- [Rao90] A. R. Rao. *A taxonomy for texture description and identification*. Springer-Verlag, New York, 1990.
- [Roc91] I. Rock and S. Palmer. L'héritage du gestaltisme. *Pour la Science*, (160):64–70, Février 1991.
- [Ros66] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *J. Assoc. Comput. Mach.*, 13(4):471–494, 1966.
- [Ros82] A. Rosenfeld and A.C. Kak. *Digital picture processing*, volume 1 and 2. Academic Press, New York, 1982. second edition.
- [Ros86] A. Rosenfeld. Some pyramid techniques for image segmentation. Technical Report CSTR-1664, Center for Automation Research, University of Maryland, May 1986.
- [Ros87] A. Rosenfeld. Recognizing unexpected objects: a proposed approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(1):71–84, April 1987.
- [S.84] S. and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [Sal93] M. Salotti and C. Garbay. Cooperation between edge detection and region growing: the problem of control. In *Proc. Int. IP:TA*, pages 95–98, San Remo, Italy, june 1993.
- [Sam80] H. Samet. Region representation: quadtrees from binary arrays. *Computer Graphics and Image Processing*, 13:88–93, 1980.
- [Sch93] M. Schmitt and J. Mattioli. *Morphologie mathématique*. Paris, 1993.
- [She86] J. Shen and S. Castan. An optimal linear operator for edge detection. In *Proceedings CVPR86*, pages 109–114, Miami, 1986.
- [She91] C.A. Sher and A. Rosenfeld. Pyramid cluster detection and delineation by consensus. *Pattern Recognition Letters*, 12:477–482, August 1991.
- [SM88] Philippe Saint-Marc and Gérard Médioni. Adaptive smoothing for feature extraction. In *Image understanding workshop*, pages 1100–1113, 6-8 April 1988.
- [Spa91] M. Spann. Figure/ground separation using stochastic pyramid relinking. *Pattern Recognition*, 24(10):993–1002, August 1991.
- [Spo87] K.T. Spoehr and S.W. Lehmkuhle. *Visual information processing*. Freeman and Company Ed., San Francisco, 1987.

- [ST75] T. Pavlidis S. Tanimoto. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4:104–119, 1975.
- [Ste78] D.F. Stevens. *Patterns in nature*. Little Brown Ed., Boston, 1978.
- [Suk83] M. Suk and T.H. Cho. Segmentation of images using minimum spanning trees. In *Proc. of SPIE, The International Society for Optical engineering*, volume 397, pages 180–185, Genova, Switzerland, April 19–22 1983.
- [Til88] J.C. Tilton. Image segmentation by iterative parallel region growing with applications to data compression and image analysis. In *Frontiers of Massively Parallel Computation*, pages 357–361, Fairfax Virginia, 1988.
- [Toe74] A. Toet. A morphological pyramidal image decomposition. *Pattern Recognition Letters*, 9:255–261, 1974.
- [Tuc90] M. Tuceryan and A.K Jain. Texture segmentation using Voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:211–216, 1990.
- [Uhr87] L. Uhr. Highly parallel, hierarchical, recognition cone perceptual structures. In Leonard Uhr, editor, *Parallel Computer Vision*, pages 249–292. Academic Press, Inc, Orlando, Florida, 1987.
- [Wil93] D. Willersinn and W.G. Kropatsch. Preserving object topology in the irregular curve pyramid. In Poepl, editor, *Mustererkennung 1993*, pages 655–662, Lübeck, September 1993. Springer.
- [Wil95] D. Willersinn. *Dual irregular pyramids*. PhD thesis, Technische Universität Wien, Institut für Automation, 1995.
- [Wit83] A.P. Witkin. Scale-space filtering. In *Proc. of the 7th International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, 1983.
- [Wro87] B. Wrobel and O. Monga. Segmentation d’images naturelles : coopération entre un détecteur-contour et un détecteur-région. In *Actes du 11^{ème} Colloque GRETSI*, Nice, France, juin 1987.
- [Zer90] J. Zerubia and R. Chellapa. Mean field annealing for edge detection and image restoration. In *proc. EUSIPCO*, Barcelona, Spain, September 1990.
- [Zuc76] S.W. Zucker. Survey region growing: childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.

Index

4-voisinage, 59

8-voisinage, 59

Agrégation de pixels, 40

algorithme

de décimation de Luby, 66

de décimation de Meer et Connelly,
67

de détection stochastique de contours,
101

de la pyramide Gaussienne, 15

de la pyramide irrégulière, 62

de segmentation par Voronoï, 25

de transmission pyramidale, 17

général de décimation, 66

top-down

itératif, 115

récuratif, 115

analyse d'image, 2

apex, 55

arête floue, 31

arbre, 111

d'attachement, 74, 76

quaternaire, 20

arbre couvrant minimal, 30

arc, 56

array processor, 13

attribut, 36, 50

Bottom-up, 19

bulle, 49

Côté, 32

carte

de champs récepteurs, 60

de contours, 50, 94

de normales, 50

cellule, 9

champ de Markov, 15, 46

champ récepteur, 9, 27, 28, 60, 68

définition, 54

classification, 39

codage, 14

entropique, 17

complexité algorithmique, 12

composante connexe, 20, 21, 29, 123

reconstruction, 60

compression d'image, 17

Connection Machine, 13

contours actifs, 48

convergence, 80

convolution, 43, 45

masque, 14

noyau, 14

impair, 14

pair, 14

produit, 14

coopération

dynamique, 51

par contrainte, 50

semi-dynamique, 51

coût de fusion, 40

critère

de similarité, 70

d'unification, 118

croissance de région, 39, 50

contrainte, 50

Décimation

étendue, 117

adaptative, 68

algorithme général, 65

- convergence, 61
- d'ordre étendu, 117
- par réseaux de Hopfield, 68
- stochastique
 - de Luby, 65
 - de Meer et Connelly, 66
- détection
 - d'objet, 11, 28
 - de points brillants, 12
- degré, 32
- Delaunay, 25
- diagramme de Voronoï, 23, 129
 - définition, 23
 - propriétés, 23
- dilatation, 45
- division, 20
- division-fusion, 50
- déclivité, 99
- dérivée directionnelle, 43

E

- chantillonnage, 28
- effets de bord, 28
- EGPA, 13
- élément de contour, 46, 92, 94
- élément de région, 94
- élément structurant, 19, 45
- énergie, 18, 28, 46, 48, 68
 - d'interaction, 49
 - de l'erreur quadratique, 47
 - de lissage, 47
 - de membrane, 46
 - image, 48
 - interne, 48
- ensemble absorbant, 73
- ensemble dominant
 - définition, 58
 - minimal, 58
 - minimum, 58
- érosion, 41, 45
- étendue, 45
- étoile, 74
- Expand, 15
- extraction de primitives, 11
- extremum local, 67

F

- face, 32
- facteur de forme, 113
- feuille, 28
- fil, 11, 12, 14, 27
- filtre
 - adaptatif, 19, 45
 - non linéaire, 19
 - optimal, 45
- forêt, 74
- formalisme de Zucker, 38
- fovéa, 8, 26
- frontière, 96
- fusion, 21, 40
 - empêchée, 92, 104
 - imposée, 92, 104

G

- DR TDSI, 135
- germe, 23, 40, 50, 52, 125
- GNC, 47
- gradient, 43
 - morphologique, 45
- graphe
 - contraction, 57
 - d'équivalence, 119
 - d'adjacence, 29, 40, 50, 59, 68
 - construction, 63
 - d'attachement, 74, 76
 - décimation, 57
 - définition, 56
 - de Delaunay, 23
 - de similarité, 29
 - non orienté, 69
 - orienté, 71
 - dual, 32
 - homogène, 119
 - non orienté, 56
 - orienté, 56
 - planaire, 23
 - définition, 56
 - pondéré, 57
 - pyramide de, 29
 - valué, 56

H

- omogénéité locale, 118

hystérésis, 43

I

Image
 binaire, 16, 20, 29
 de consensus, 76
 de profondeur, 50, 103
 images test, 135

L

Logique floue, 31

M

Machines parallèles, 12
 marqueur, 41
 masque
 de Kirsh, 43
 de Prewitt, 43
 de Sobel, 43
 Laplacien, 44
 mean field annealing, 47
 minimum spanning tree, 30
 morphologie mathématique, 40, 45
 moyenne, 11
 multi-échelle, 33
 multirésolution, 3, 7, 14, 30, 33, 107
 définition, 108
 principes, 8

N

N-tree, 30
 neurone, 26, 36
 norme ANSI, 128
 norme du gradient, 43
 noyau, 73

O

Ondelettes, 15
 opérateur d'intérêt, 68, 124
 ordre
 de Morton, 20
 de Peano-Hilbert, 20

P

Père, 12, 14, 27
 PAPIA, 13
 paramètre d'échelle, 18, 113
 partition, 38
 point de contraste, 50
 polygone de Voronoï, 23

post-traitement, 88
 pré-segmentation, 50
 processus de ligne, 46
 processus intensité, 46
 Pyramid Vision Machine, 13
 pyramide
 à base arborescente, 30
 à base hexagonale, 26
 à base locale, 120
 à recouvrements, 27
 adaptative, 30, 68
 de contours, 31
 de courbes, 26
 de graphe, 29
 de Peleg, 28
 de régions, 29
 double, 9
 duale, 31
 dynamique, 18, 79
 floue, 31
 Gaussienne, 14
 irrégulière, 29
 principes, 54
 Laplacienne, 16
 liée, 27
 liée stochastique, 28
 quadtree, 21
 quaternaire, 11, 21, 26
 reliée, 28
 souple, 30
 stochastique, 29

Q

principe, 20

R

Réseau de Hopfield, 68
 résolution, 20
 définition, 108
 racine, 20, 27, 28, 82, 111
 range image, 103
 rapport de contraction, 82
 recuit par champ moyen, 47
 recuit simulé, 28
 Reduce, 15

réalisation, 82
réseau T. Node, 13
résolution, 28

Séquence d'images, 18
scale-space, 18
segmentation, 2, 29, 36
 coopérative, 48
 définition, 36, 38
 par contours, 41
 par régions, 37
 probable, 76

seuil
 global, 70
 hystérésis, 43

seuillage, 39
site de contour, 46
snake, 48

sommet
 de Delaunay, 23
 de Voronoï, 23
 flou, 31
 unificateur, 120, 121

SPHINX, 13

split and merge, 20
 contraint, 50

stable
 définition, 58
 maximal, 64
 définition, 58

Substract, 16

sur-segmentation, 45

T_{aux}
 de décimation, 82, 125
 de réduction, 26

temps de traitement, 80

tesselle, 130

texture
 définition, 128
 orientée, 129

tirage aléatoire, 66, 68, 75, 85

token, 129

top-down, 30

 multirésolution, 112
 principe, 20
topologie, 32
Transputers, 13
triangulation de Delaunay, 129

Variance, 11, 21
Voronoi, 129
 structure, 23

Zone d'intérêt, 51



Contribution des pyramides irrégulières en segmentation d'images multirésolution

Pascal Bertolino

► To cite this version:

Pascal Bertolino. Contribution des pyramides irrégulières en segmentation d'images multirésolution. Human-Computer Interaction [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 1995. French. <tel-00005028>

HAL Id: tel-00005028

<https://tel.archives-ouvertes.fr/tel-00005028>

Submitted on 24 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.